AD 295 696

295 696

63-2-3

6-90-62-83 • SEPTEMBER 1962

6-90-62-83

TECHNICAL REPORT: MATHEMATICS

METHODS OF INCREASING REDUNDANCY IN A CLASS OF PATTERN RECOGNITION SYNTHESIS ALGORITHMS: AN EXPERIMENTAL STUDY

WORK CARRIED OUT AS PART OF THE LOCKHEED INDEPENDENT RESEARCH PROGRAM.

6-90-62-83 • SEPTEMBER 1962

6-90-62-83

TECHNICAL REPORT: MATHEMATICS

METHODS OF INCREASING REDUNDANCY IN A CLASS OF PATTERN RECOGNITION SYNTHESIS ALGORITHMS: AN EXPERIMENTAL STUDY

by

R.L. MATTSON

O. FIRSCHEIN

M. FISCHLER

WORK CARRIED OUT AS PART OF THE LOCKHEED INDEPENDENT RESEARCH PROGRAM

Lockheed

MISSILES & SPACE COMPANY

A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION

SUNNYVALE, CALIFORNIA

NOTICE

FOREWORD

This report is the fifth in a series of publications dealing with various aspects of pattern recognition. The first two publications discussed an approach to pattern recognition using linear threshold devices, one dealing with problems of extracting significant characteristics from the pattern, the other with the design of the decision making device which operates on these characteristics to perform classification. An overall experimental system was described in the third report, while methods for coding the significant characteristics were discussed in the fourth report.

The present paper investigates an algorithm which synthesizes a network of threshold elements to perform pattern recognition. This algorithm was originally designed to simplify decision-making networks by elimination of redundant information. The pattern recognition experiments presented in this report are designed to do the opposite, i.e., to find out how the generalization abilities of the algorithm might be improved.

This investigation will be continued to determine the effect of other variations of the synthesis algorithm on pattern recognition.

ABSTRACT

This report investigates an algorithm which synethsizes a network of threshold elements designed to perform pattern recognition. Pattern recognition experiments were conducted to determine what the generalization abilities of the algorithm were, and how these could be improved.

Basic concepts of feature space geometry, threshold devices and networks in their application to pattern recognition are discussed. Four methods for developing a synthesis algorithm are presented. The fourth, the Multiple Threshold per Class with Minimal Weights method, is the basis for the algorithm used in the investigation. The technique of determining weight and threshold is presented.

A description of the Pattern Information Processor, a special purpose computer used extensively in the investigations, precedes discussion of the experiments. The purpose of the experiments was to devise methods for increasing the generalization ability of the synthesis algorithm. The two basic approaches were:

(1) Modification of data presented to the algorithm. This can be accomplished by:

- Increasing the size of the organizing set
- Increasing the noise in the organizing set

(2) Revision of the basic algorithm by starting with non-zero weights. It was concluded that for a feature-word space consisting of ideal characters with additive noise, best results can be obtained with an algorithm which used initial correlated weights.

ACKNOWLEDGMENTS

The authors wish to thank Dr. R. I. Tanaka, Senior Member, Electronics Sciences Laboratory, for his encouragement and support during the course of the experiments: E. A. Poe for the checkout of PIP; J. Sharp for laboratory services; and J. B. Bridges for the computer programming required for the experiments.

CONTENTS

ILLUSTRATIONS

Section 1
INTRODUCTION

This report , the fifth in a series of publications dealing with various aspects of pattern recognition (Refs. 1 – 4), presents several methods for synthesizing a decision-making system designed to perform pattern recognition. These methods range from common correlation approaches to the more sophisticated threshold element network approach. Each approach is described and the limitations and advantages are discussed. The threshold element network approach is investigated more extensively because it is easy and economical to mechanize.

Syntheses algorithms may be developed from several points of view. For example, it may be desirable to synthesize minimal device networks, minimal weight devices, or networks having a certain symmetry. For pattern recognition, however, the important attribute of any syntheses algorithm is that it have "generalization ability," i.e., that the network should be able to properly classify patterns which are not members of the organizing set.

A pattern consists of a number of component factors, none of which has individual significance, and may, or may not be present, depending on a given situation. The loss or modification of a small number of these factors does not in any way affect the meaning of the pattern; that is, the pattern remains unchanged, even though some of its component factors have been modified or destroyed. Certain factors, then, may be considered to be redundant, for in an optimum situation, the pattern could still be recognized without these factors. If a decision-making system bases its decision on only a few factors, it is not using the natural redundancy in the pattern. Should some factors in a pattern be missing, misrecognition could result, even though enough information is present for correct identification. The problem, therefore, is to develop an algorithm which designs a decision-making system capable of using all redundant information in the pattern.

The algorithm investigated in this report synthesizes a network of threshold elements that perform pattern recognition. It was originally designed to produce simple decision networks which use the minimum amount of information in a pattern. The pattern-recognition experiments presented in Section 6 were performed to determine methods of forcing the basic design algorithm to form decision networks which use all the information in the pattern, even though some of this information may be redundant. Section 2 defines and briefly discusses the basic concepts involved. Section 3 presents four different types of syntheses algorithms; Section 4 describes the minimum redundancy algorithm used to design networks of threshold elements for performing pattern recognition. Section 5 describes the PIP computer which made possible the extensive investigation of syntheses algorithms and decision making systems.

Section 2
BASIC CONCEPTS

2.1 FEATURE-SPACE GEOMETRY

If the characteristics or features of a pattern are viewed as the axes of a Euclidean space, an intuitive picture of the classification procedures discussed in Section 3 can be obtained. For example, Fig. 2-1 shows the feature words for a set of patterns, where "0" in a position of a word designates the absence of a feature, and "1" designates the presence of a feature. Associated with each feature word is a binary tag which gives the classification of the pattern. The patterns shown have been arbitrarily divided into three classes, α, β and γ, with the binary tags 10, 01, and 00, respectively. Each column, I and II, of the desired classification can be considered as determining a partition of the set of patterns. Thus, if a device is synthesized which performs the partitioning for each column, then the desired classification can be obtained. The features connected, rectilinear, and containing a loop, can be considered as forming a three-dimensional space as shown in Fig. 2-2, and classification can be regarded as the partitioning of this space by one or more planes. Thus, in Fig. 2-2, the dividing plane shown maps feature words for nontriangular patterns on one side of the plane, while the feature word for the triangular pattern is mapped on the other side of the plane. This classification concept, which can be generalized by considering the space to be a hyperspace partitioned by hyperplanes, not only offers conceptual advantages, but also facilitates ready implementation by means of threshold devices.

2.2 THRESHOLD DEVICES

A schematic diagram of a threshold device is shown in Fig. 2-3. The inputs to this device are binary, "0" and "1;" the weights, w_1, and the threshold, T, are

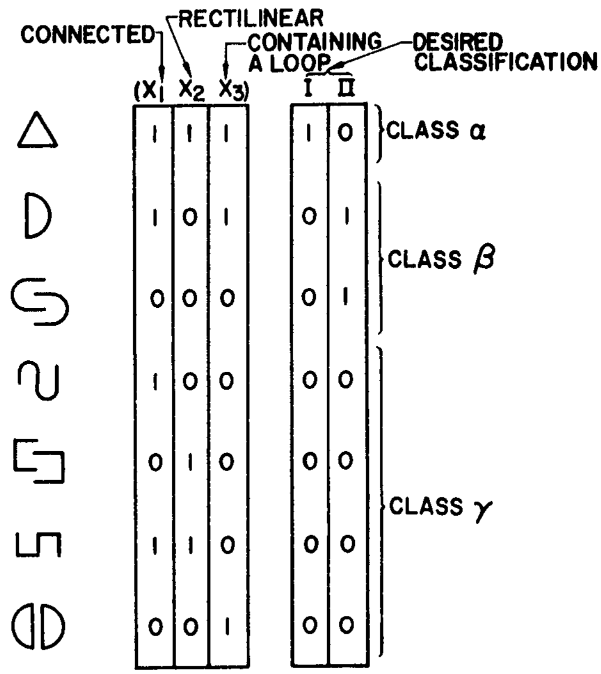


Fig. 2-1 Feature Words for Various Patterns

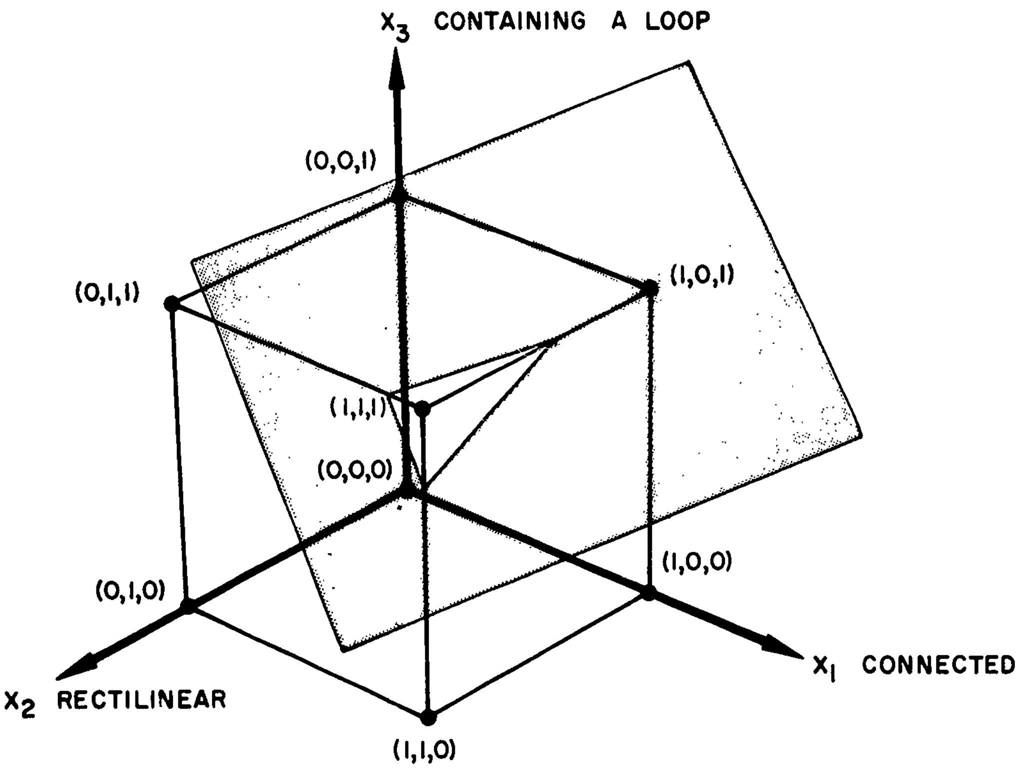


Fig. 2-2 Feature Words Viewed as Vertices of a Cube

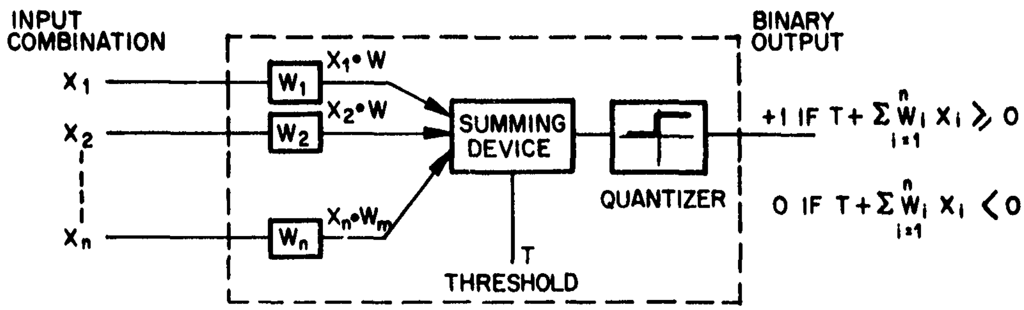


Fig. 2-3 Representation of Threshold Device

integers that can assume either positive or negative values. The output of this device is 1 if

$$T + \sum_{i=1}^{n} x_i w_i \geq 0 \tag{2.1}$$

otherwise, it is "0." The dividing condition between the output being 0 or 1 is given by

$$T + \sum_{i=1}^{n} x_i w_i = 0 \tag{2.2}$$

Equation (2.2) represents a hyperplane in the x_i space. Viewing the x_i combinations as vertices of a hypercube, one can say that the hyperplane partitions the hypercube so that all points on one side of the plane are mapped as a 1, while the points on the other side of the hyperplane are mapped as "0." The w_i values control the slope of the hyperplane, and the threshold, T, controls the position of the plane in the space. Classification, then, consists of finding the parameters which properly orient the partitioning plane.

2.3 NETWORKS OF THRESHOLD DEVICES

A set of feature words and the desired classifications for the patterns of Fig. 2-1 are given in Fig. 2-4; the threshold network for classifying these words is shown in Fig. 2-5. It will be noted that Classification Column I required only a single threshold device, but that Column II required a network of two devices. Column I is said to determine a "separable" set of feature words. The criteria for separable and non-separable feature words have been extensively investigated in recent literature (Refs. 5 and 6).

In the example shown, an OR gate has been used to combine the outputs of the threshold devices. Other mechanizations are of course possible; an alternative method of combining the outputs is discussed in Section 4.

2.4 PATTERN RECOGNITION APPLICATIONS

In problems where all the feature words and their desired classification are known, a network of threshold devices can readily be used to perform exact word classification. For pattern recognition problems, however, the number of possible variations of a pattern is so large that it becomes impractical to map all possible feature words. The hyperplanes are therefore positioned by using only a representative set of words; the representative set determines the mapping criterion. The pattern recognition process is represented schematically in Fig. 2-6. It should be noted that this is a general decision process, and not limited to any specific problem. In fact, any problem with typical characteristics which can be expressed in a binary feature word and which can be reduced to a representative set of words - each tagged with a desired classification - can be handled by this method.

However, any such problem may be complicated by the following aspects:

- Choice of significant characteristics to be used in making up the feature word
- Coding of these characteristics to form a feature space
- Mechanization in the construction of the feature word

DESIRED CLASSIFICATION

FEATURE WORDS X_1	X_2	X_3	I	II	
1	1	1	1	0	CLASS α
0	0	0	0	1	CLASS β
1	0	1	0	1	
1	0	0	0	0	CLASS γ
0	1	0	0	0	
1	1	0	0	0	
0	0	1	0	0	

Fig. 2-4 Set of Feature Words and Desired Classifications

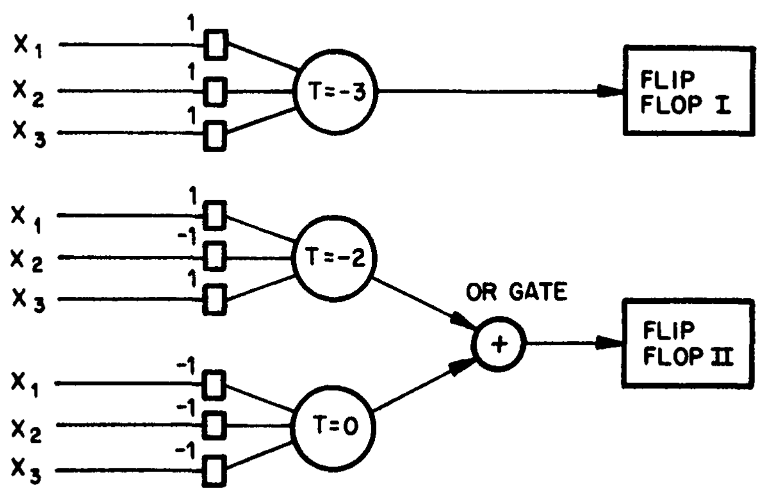


Fig. 2-5 Threshold Network for Classifying Feature Words

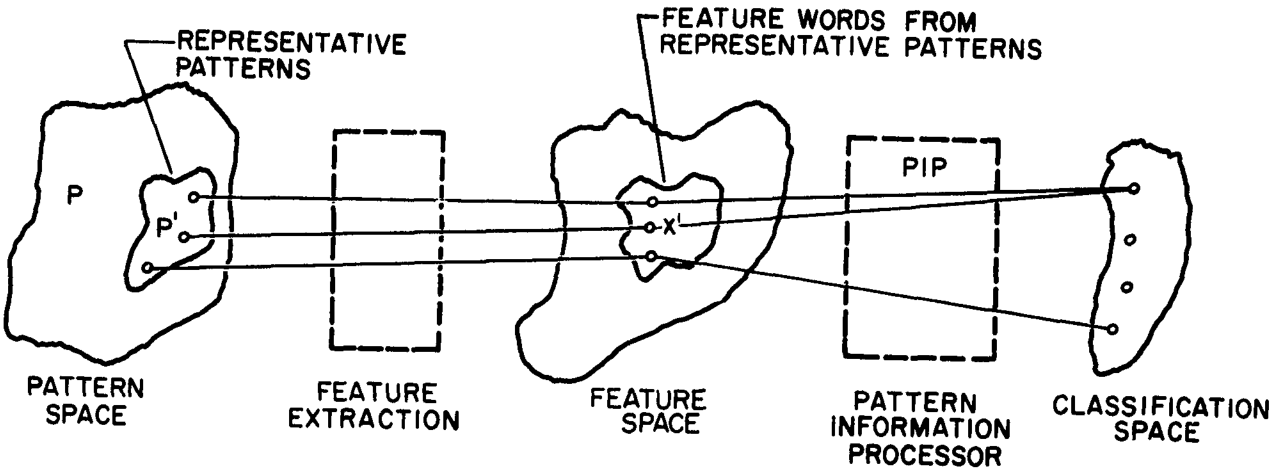


Fig. 2-6 The Pattern Recognition Process

Figure 2-7 indicates the difficulty of classification in a space whose feature words are not relevant to the desired classification. The classes α, β, and γ are spread throughout the space. This is in contrast with the situation in Fig. 2-8, where the classes can easily be separated by hyperplanes.

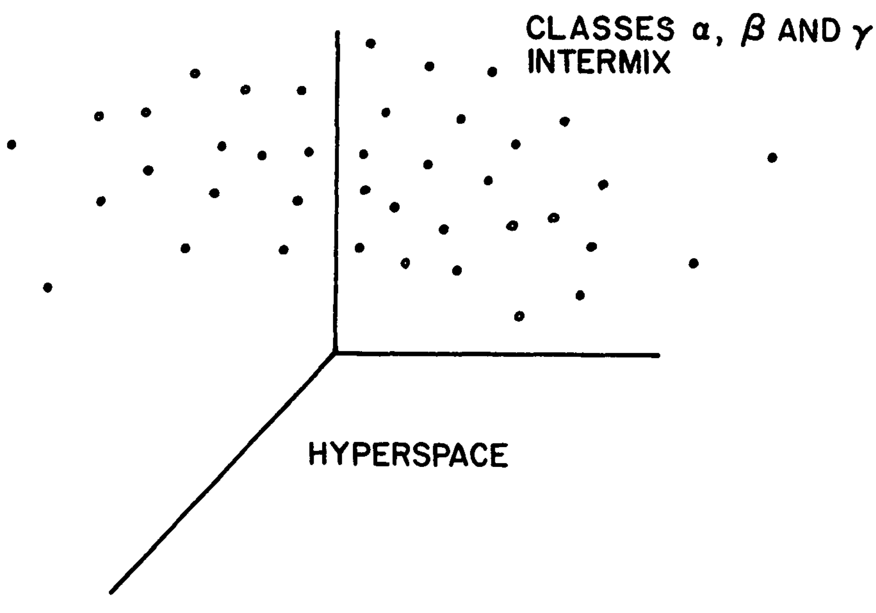


Fig. 2-7 Location of Feature Words in a "Poorly Chosen" Feature Space

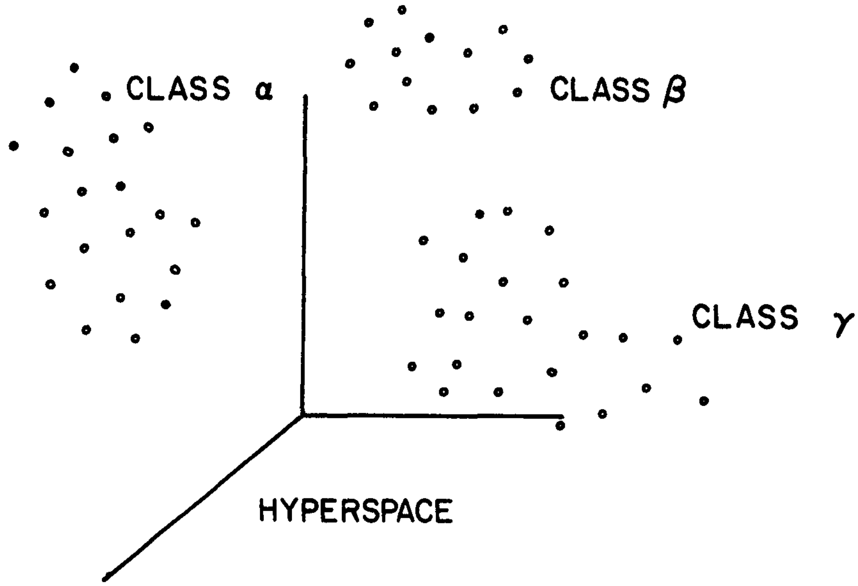


Fig. 2-8 Location of Feature Words in a "Well Chosen" Feature Space

Section 3
THRESHOLD SYNTHESIS ALGORITHMS

3.1 INTRODUCTION

Synthesis algorithms may be developed from several points of view (Refs. 7 and 8). It may be desirable to synthesize minimal device networks, minimal weight devices, or networks having a certain symmetry. For the pattern recognition problem, however, the important attribute of any synthesis algorithm is that it have "generalization" ability. Thus, once the threshold parameters have been determined,the network should be able to properly classify patterns which are not members of the organizing set.

This section describes the development of a synthesis algorithm. It starts with a simple correlation scheme chosen from the many possible types of correlation to demonstrate the choices available in the design of a synthesis algorithm.

3.2 METHODS FOR DEVELOPING SYNTHESIS ALGORITHMS

Four classification schemes are discussed in this section:

- The Maximum Correlation Score Method (MCS)
- The Single Threshold per Class Method (STC)
- The Multiple Threshold per Class (Redundant Weight) Method (MTCRW)
- The Multiple Threshold per Class (Minimal Weight) Method (MTCMW)

A discussion of the application of these methods is given in Section 3.4. Feature words using ±1 designation rather than 0, 1 are used.

3.2.1 Maximum Correlation Score Method

Given a set of typical feature words which are divided into classes, it is possible to classify an unknown feature word by using the following simple correlation scheme. Take the inner product (dot product) of the unknown word with each feature word of each class, and average over each class the correlation numbers obtained for that class. The unknown word is then placed in the class for which the highest average correlation number was obtained. This approach will be called "Maximum Correlation Score" classification.

Because of the linear operations involved in obtaining the correlation number, the feature words defining a given class can be added together vectorially and each component of the resultant vector can be divided by the number of words to give a "weight" vector. The correlation number between an unknown feature word and a given class can now be obtained by taking the inner product between this feature word and the weight vector for this class. The weight vector may be said to "represent" the class.

3.2.2 Single Threshold Per Class Method

Instead of recording and comparing correlation numbers for each class in order to classify an unknown feature word, we can simplify the decision rule by assigning the unknown feature word to a particular class if its correlation number exceeds a single fixed threshold number associated with that class. It should be noted that this simplification does not imply improvement from a classification standpoint; it is merely useful in clarifying the discussion of the final algorithm. When the STC is used, an unknown feature word is rejected if it is assigned to two or more classes, or if it is not assigned to any class. If the thresholds are set too high or too low, a high rejection rate will result, but misclassification will be less likely with higher settings.

3.2.3 Multiple Threshold Per Class With Redundant Weights

Because in the previous method only one threshold number is used for each class, it is possible that a typical feature word in a class will fall below the threshold number

for that class. This situation may be remedied by retaining the weight vectors and by using more than one threshold for each class (Fig. 3-1). If two classes have typical

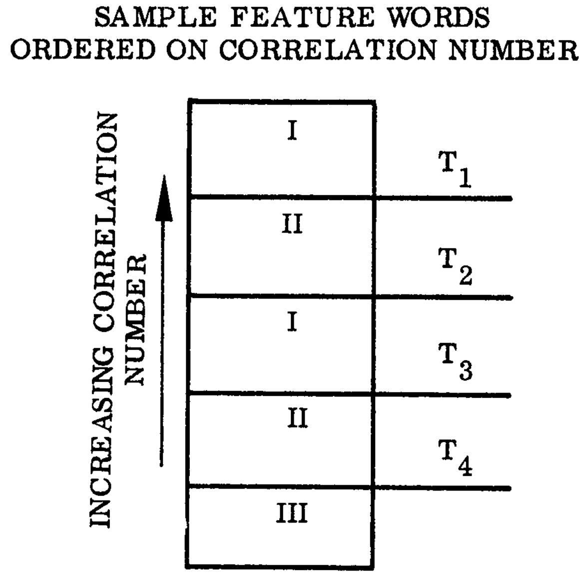


Fig. 3-1 Multiple Threshold Classification, Showing Class Membership Between Threshold Levels for a Particular Weight Vector

feature words which cannot be separated by adding additional thresholds, then it is necessary to modify the weight vectors until separation of these words is obtained. When multiple threshold classification is used, the unknown feature word is tested to determine between which thresholds the word falls.

This method can be thought of as a "switching theory approach," since the set of feature words can be considered to be an incompletely specified table of combinations whose entries have been assigned function values. A switching network which mechanizes this table of combinations will also classify the missing entries, and thus be equivalent to a complete decision procedure.

Viewed in geometrical terms, the decision surfaces are hyperplanes in the feature space. The hyperplane is moved through the unit n-cube (which has vertices of carefully selected typical feature words) in such a way that only vertices from one class fall on the positive-side of the hyperplane. When additional movement is impossible

without including points from another class, then the position of the hyperplane is fixed and the typical vertices on the positive half of the hyperplane define a subclass.

3.2.4 Multiple Threshold Per Class With Minimal Weights Method

This method is similar to the method described in Section 3.2.3, except that a further logic design approach is taken by setting elements in the typical weight vectors to zero whenever possible. Since fewer weight elements are used in calculating the correlation numbers, the decision mechanization is simplified. This approach, however, causes recognition problems as is noted in the following section.

3.3 CLASSIFICATION APPROACHES

3.3.1 Single Threshold Per Class Method

When the designer sets up a coding scheme for transforming from pattern space to feature space, it is desirable that the feature words defining a given class have a higher degree of correlation among themselves than with feature words from other classes. If the designer succeeds in this objective, it is possible to use the MCS or the STC methods. If it is important that rejection, rather than misclassification be obtained for a decision situation, the STC method is preferable.

3.3.2 Multiple Threshold Per Class Method

When the use of the above method results in rejection or misclassification of most of the typical feature words for a class, the MTC approach becomes necessary, since it appears advantageous to have each representative feature word of a class be correctly classified. If the total number of thresholds required becomes too large, then the system is impractical and a new transformation scheme is required. On the other hand, if too few thresholds are used, the misclassification and reject rate will be too high.

3.3.3 Comparison of the Classification Approaches

In comparing MTC methods with the MCS and STC methods, we note that in the latter two methods only the typical feature words in the class for which the weight vector is being constructed are used in the construction. In particular, if a new class is defined after the correlation weight vectors have been computed, no changes in these computed weight vectors will be necessary.

In the Multiple Threshold Per Class methods, however, all the typical feature words affect the construction of the weight vector for a particular class. If a new class is defined, or even if additional typical feature words for the other classes are introduced, all the previously computed weight vectors are subject to modification. Thus, in the Multiple Threshold methods, we are attempting to distinguish between a set of clearly defined objects, while in the MCS and STC methods, we are attempting to identify objects in a given class by constructing the weight vectors which use no information about what other types of objects can occur.

The MCS and the STC methods cannot determine what is redundant information. Consequently, these methods cannot reduce redundancy. The MTCRW method makes no statement about how much redundancy is desired, while the MTCMW method takes the usual logic design approach of minimizing redundancy. The decision rule obtained when the last-mentioned method is used is probably economical to mechanize, but less effective in handling feature words for which all bit positions are important, and the relative importance of the bit positions is unknown.

Because of the above-mentioned mechanization advantage, the experiment described in Section 6.3 used an algorithm based on this method.

Section 4
WEIGHT AND THRESHOLD DETERMINATION

4.1 INTRODUCTION

The following exposition of the weight and threshold algorithm is a descriptive explanation of the technique. It will provide the foundation for a better understanding of the various modifications of the algorithm discussed in Section 6. A formal mathematical discussion of the threshold synthesis problem is given in Ref. 6.

The algorithm follows the philosophy that once a 1-mapped feature word has been separated by a set of weights and thresholds, it remains separated as these parameters are varied to pick up additional 1-mapped points. This synthesis ends as soon as all the 1-mapped points have been separated. No attempt is made to shift the threshold, or to change the orientation of the separating hyperplanes once separation has occurred. Certain programming refinements and procedures are used to keep the magnitude of the weights small and to decrease computation time. However, since these techniques are not pertinent to the theory, they are not included in the following discussion.

The computational arrangement for the algorithm is given in Fig. 4-1. The notation "d.m." is used for desired mapping (classification), and "Σ" is an abbreviation of $\sum_{i=1}^{N} w_i x_i$. The feature words are sorted according to their associated Σ column, with the highest Σ on top of the list. There are two nontrivial cases that arise as the computation proceeds:

Case 1: The first 1-mapped feature word below the threshold has a Σ equal to the Σ of the first 0-mapped word (Fig. 4-2).

Case 2: The first 1-mapped feature word below the threshold has a Σ less than the Σ of the first 0-mapped word (Fig. 4-3).

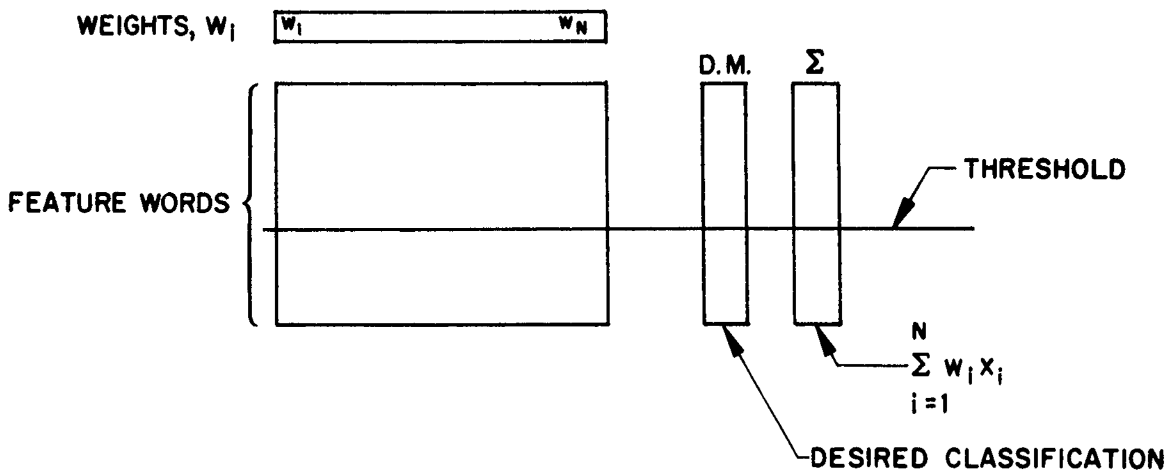


Fig. 4-1 Computational Arrangement

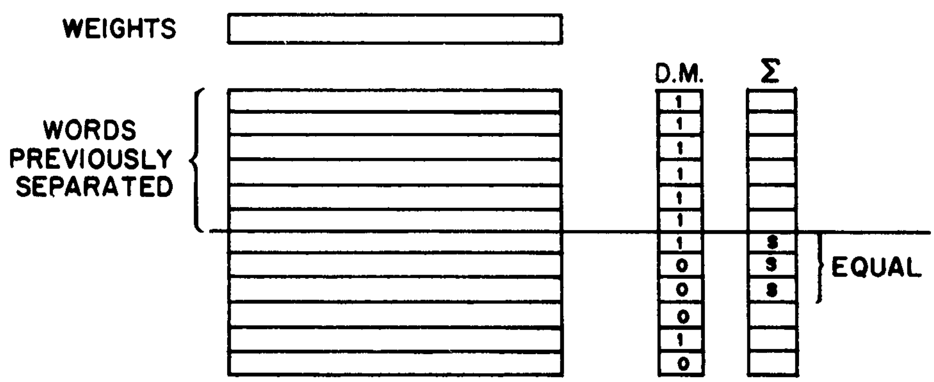


Fig. 4-2 Case 1: 1-Mapped Word With Σ Equal to Σ of O Mapped Words

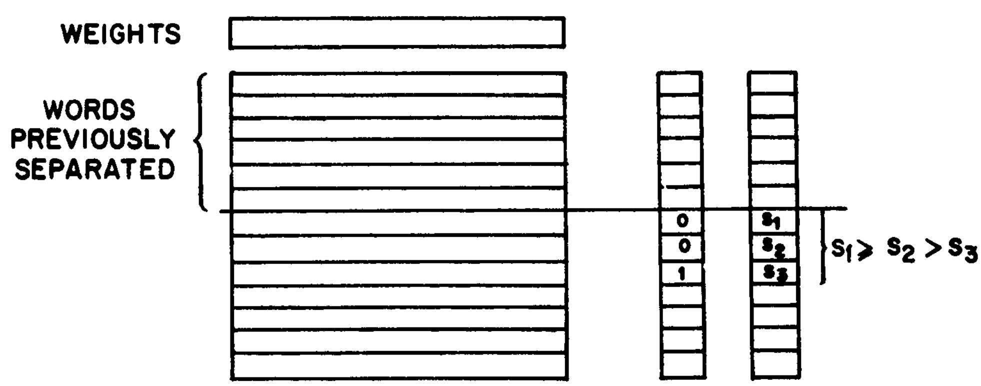


Fig. 4-3 Case 2: 1-Mapped Word With Σ Less Than Σ of O Mapped Words

(The third case, a 1-mapped word below the threshold with Σ greater than any 0-mapped word, is trivial since the threshold is simply decreased to include this word above the threshold.)

4.2 TECHNIQUE

4.2.1 Case 1

Figure 4-4 shows a 1-mapped word and several 0-mapped words which all have the same Σ. The 1-mapped word can be made to have the largest Σ by adding a +1 weight to the columns for which a "1" appears in the 1-mapped word, and adding a -1 weight to the columns for which a "0" appears in the 1-mapped word. When it is desirable to keep the weights small, the technique shown in Fig. 4-5 can be used. This procedure changes weights column by column until the 1-mapped word has the largest Σ.

The Case 1 procedure operates on a single 1-mapped word and on those 0-mapped words whose Σ equals the Σ of the 1-mapped word. The weights obtained for a subset of feature words are called "working weights." When working weights are added to the "main weights" (weights which apply to the entire set of feature words), it is possible that a previously separated 1-mapped word will now have a Σ below the threshold. Therefore, a check procedure which tests for this situation is included; if such a case arises all the main weights are doubled before the working weights are added.

4.2.2 Case 2

A set of 0-mapped words whose Σ is larger than the 1-mapped word is shown in Fig. 4-6. For this case, the columns of the array are examined to discover a situation in which the column entries for the 0-mapped words all differ from the column entry of the 1-mapped word. Figure 4-6 indicates the weight modification to be made if such a situation exists. The result of such a modification is a Case 1 situation which will insure separation of the 1-mapped word. If no columns satisfy the requirements, then an additional threshold device will be synthesized.

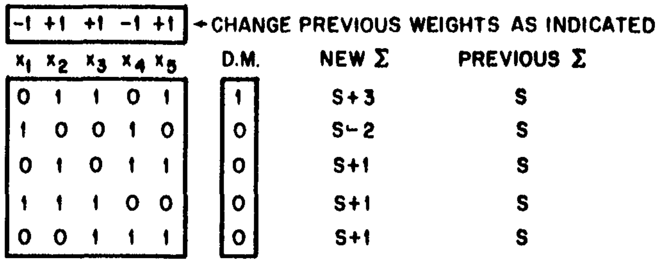


Fig. 4-4 Technique for Case 1

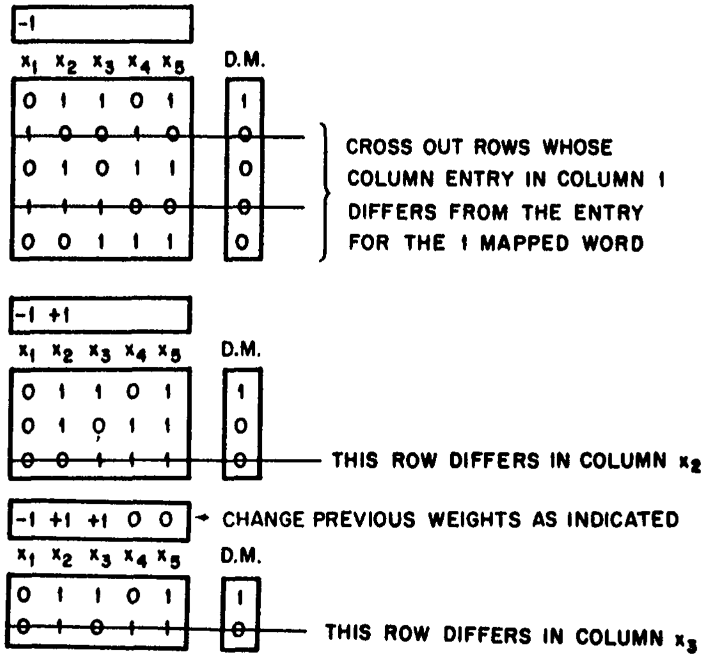


Fig. 4-5 Alternate Technique for Case 1, Weights Kept Small

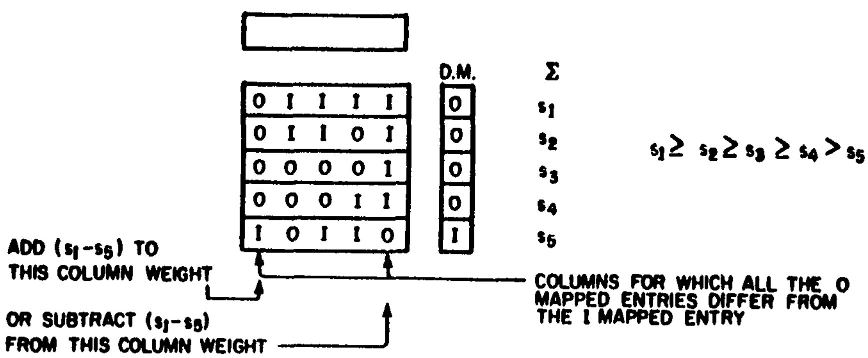


Fig. 4-6 Technique for Case 2

4.3 SYNTHESIS OF THRESHOLD NETWORKS

If the array fails the Case 2 test, a network of devices rather than one device must be formed. Figures 4-7 and 4-8 show two different approaches to the multiple-device situation. In the "parity" network procedure, the hyperplanes are determined so that once a vertex is "1" mapped by a hyperplane, it is "1" mapped by all succeeding hyperplanes. Figure 4-7 indicates vertices with a "1" desired mapping as a solid black circle. The following has been mapped:

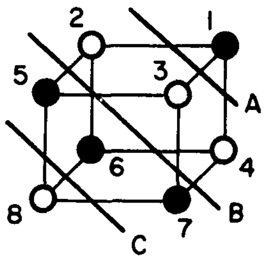


Fig. 4-7 Geometric Representation of Parity Output Procedure

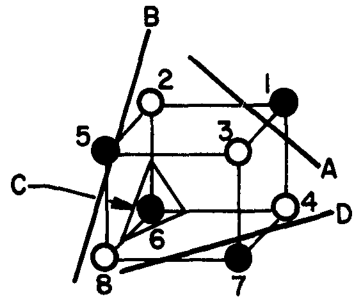


Fig. 4-8 Geometric Representation of OR Output Procedure

Point	Mapped as "1" by Planes
1	A, B, C
2, 3, 4	B, C
5, 6, 7	C
8	none

If an odd number of planes is required in a network, then any vertex which is "1" mapped by an odd number of planes has a "1" desired mapping. Thus, the correct output of a network can be obtained by noting the parity of the "1" mappings.

To use this parity procedure, the weights and threshold obtained up to this point are now stored as device A, and a new device which retains the old weights is started (Fig. 4-9). All the entries in the d.m. column below the original threshold are

complemented; the threshold is moved down to pick up the new 1-mapped points; and the Case 1 and Case 2 procedures are repeated. Whenever a Case 2 situation fails, the current weight and threshold values are stored as an additional device, and the complementing procedure is repeated until all the feature words have been separated. Outputs of the devices are combined as shown in Fig. 4-10. (The overall procedure starts with "0" weights and thresholds, thus insuring a Case 1 condition initially.)

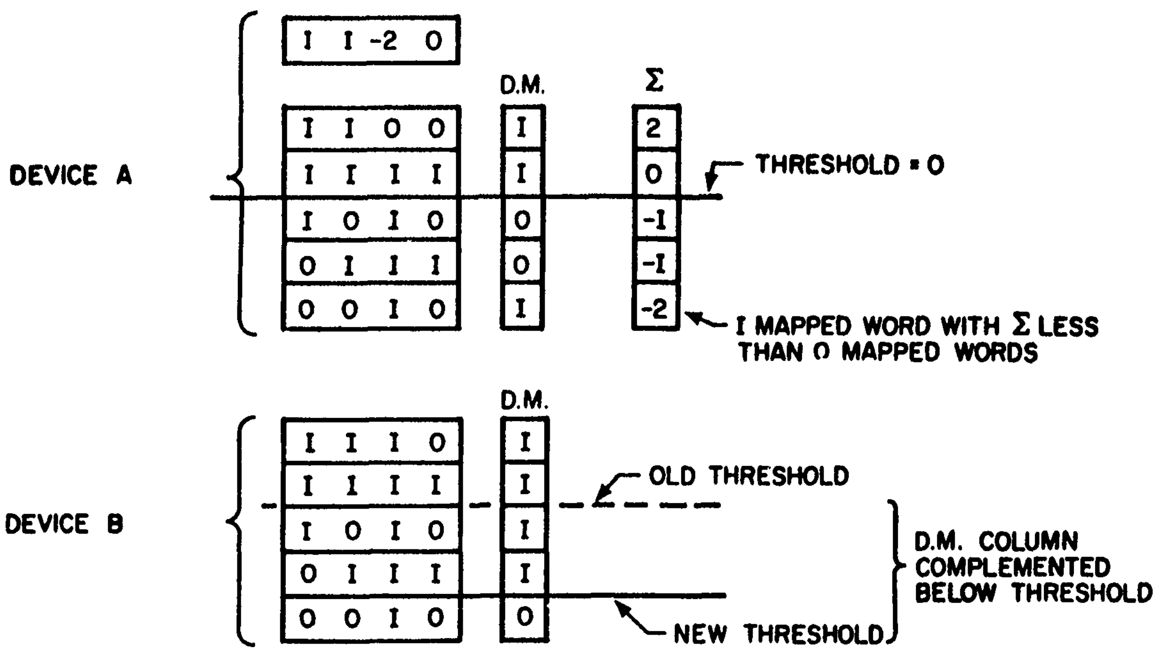


Fig. 4-9 Technique for Handling a Case 2 Failure

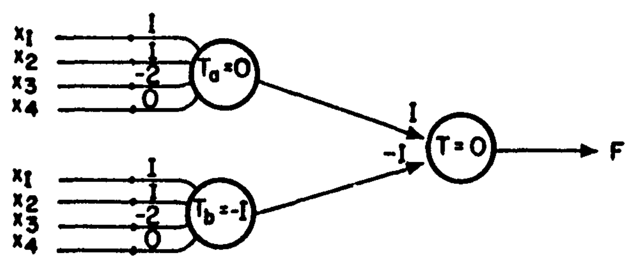


Fig. 4-10 Combining the Outputs

In the OR network procedure of Fig. 4-8, only vertices with a "1" desired mapping are "1" mapped by any hyperplane.

Point	Mapped as "1" by Plane
1	A
5	B
6	C
7	D
2, 3, 4, 8	none

The correct output of a network can be combined by using the output of each device as input to an OR gate.

When the OR output procedure is used, the 1-mapped words which have been separated are eliminated from the array; the weights and threshold are stored for the device, and the computations for the new array are started again with the weights and threshold set to 0. The OR output procedure will be tested in future experiments.

4.4 DISCUSSION OF THE ALGORITHM

Because testing of the columns starts from the left-hand side of the array, there is a tendency to use more weights in the left-hand side of the feature word. Also, the algorithm is dependent on the order of the feature words in the array. Furthermore, because the algorithm terminates immediately upon separating the 1-mapped words, it is possible that the type of generalization required for general pattern recognition will not be developed, although the organizing set is correctly separated.

Section 6 describes the methods that were devised to obtain better generalization on the part of the algorithm. It was found that the best results were obtained with an algorithm which used an initial weight setting procedure related to correlation.

Section 5
PATTERN INFORMATION PROCESSOR

5-1 DESCRIPTION OF PIP

Figure 5-1 shows the Pattern Information Processor (PIP), a special purpose digital computer which simulates a network of threshold devices. It is an excellent laboratory tool for investigating network synthesis algorithms and feature word construction methods, and was used extensively in the experiments discussed in Section 6.

The PIP uses a magnetic drum memory to store the weight and threshold values needed to simulate a variety of one-layer threshold networks. Each PIP threshold element has 6 m input lines, where m can equal 1, 2, ... , 85, and can either set or reset any one of ten output flip-flops. Thus, very versatile networks can be simulated with from 6 to 510 inputs and from 1 to 10 outputs. A detailed description of PIP appears in Ref. 9.

The number of elements that can be simulated with PIP depends on the number of inputs to each element. Approximately 8,500 9-bit weights can be stored on the drum memory. A network of D devices, each having N inputs can be simulated, where the approximate relation $N \cdot D \sim 8,500$ is satisfied. For example, 130 60-input elements can be stored on the drum while only 27 324-input elements can be used.

A threshold network using 320 bit devices is shown in Fig. 5-2, and the mechanization on the PIP drum memory is shown in Fig. 5-3. Since only five of the allowable 28 devices have been used, the drum is only partially filled. The weights, which are eight bits plus sign are stored in the drum memory as shown in Fig. 5-4. Each set of weights is followed by an eighteen-bit threshold and a nine-bit "program" word. The program word determines which output flip-flop the threshold device will control, as well as whether the flip-flop will be set or reset when the threshold device fires.

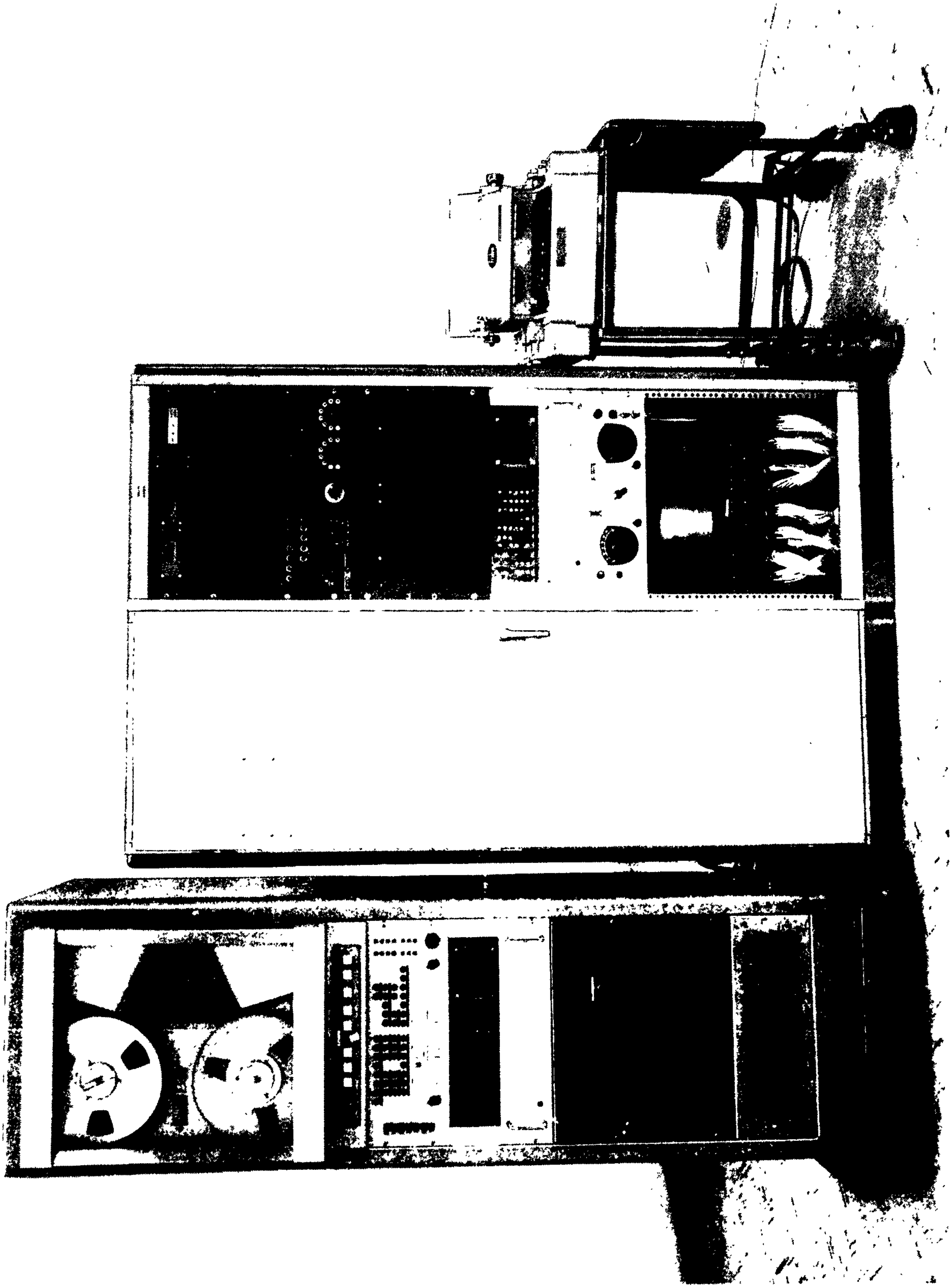


Fig. 5-1 The Pattern Information Processor

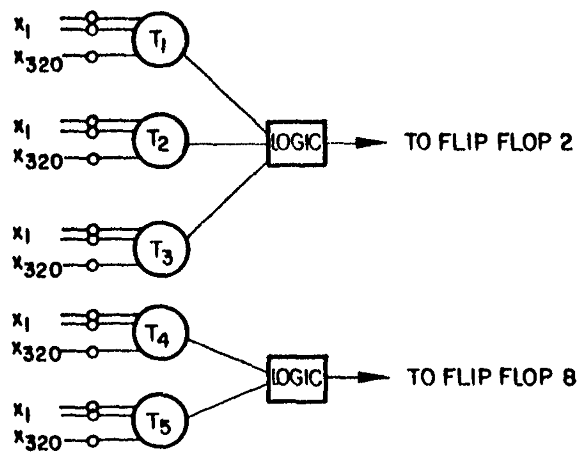


Fig. 5-2 Threshold Network

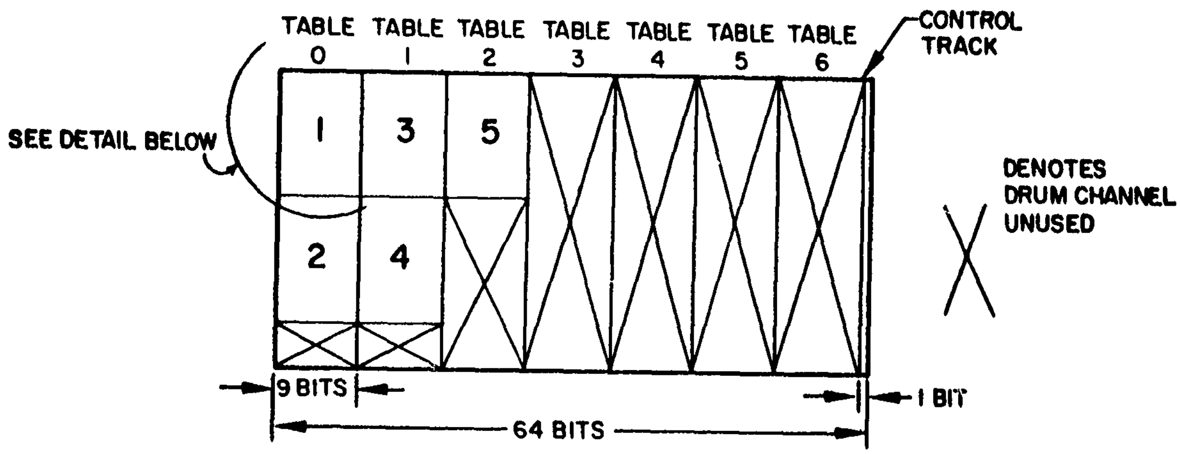


Fig. 5-3 Layout of PIP Drum for Threshold Network of Fig. 5-2

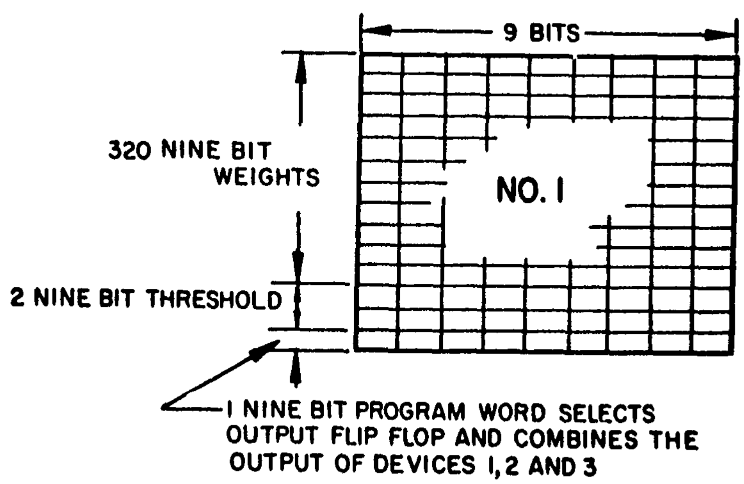


Fig. 5-4 Detail of Drum for Device No. 1

5.2 LOGICAL DESIGN OF PIP

The logical design of PIP is shown in schematic form in Fig. 5-5. A feature word is read from the tape and stored in a buffer unit. The feature word is then shifted out of the buffer and the weight corresponding to the current bit position is read from the drum. This weight is either added to an accumulator or not, depending on whether the current bit of the feature word is a "1" or "0". After all the bits of the feature word have been shifted from the buffer, the negative of the threshold value is added to the accumulator. The resulting sign of the accumulator determines whether the output of that threshold device has fired or not for that feature word. The feature word is recirculated in the data buffer until all the threshold devices stored on the drum have used this word as input. The configuration of the ten flip-flops then represents the PIP classification for that feature word.

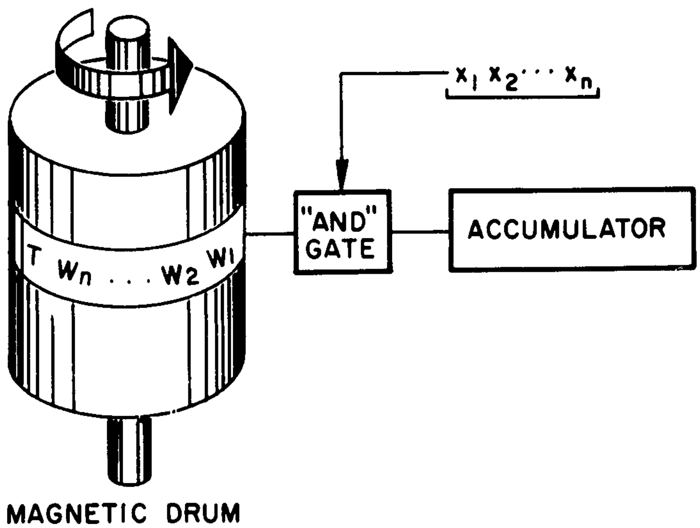


Fig. 5-5 Digital Mechanization

If the feature word was tagged originally with a desired classification, any discrepancy between the PIP classification and the desired classification can be indicated on an error counter or on the output typewriter. Errors can be indicated comparing all ten

flip-flops or any one flip-flop, as determined by selection switches. This option has been provided so that the following two types of experiments can be run:

- One treats each column as an independent function.
- The other uses code combinations involving all ten flip-flops.

5.3 OPERATIONAL ASPECTS OF PIP

To perform a recognition experiment on the PIP, the drum is loaded by means of the magnetic tape input. (For certain studies, it is necessary to change only certain threshold devices. In such cases, only certain channels of the drum must be reloaded.) Option switches are then set which select either the OR or the parity network output mode, as well as the desired mapping column to be compared on the error counter. A print mode switch determines whether typing is suppressed, whether only the errors are printed,or whether all results are printed.

The high processing speed of the PIP makes it quite practical to analyze thousands of feature words for each experiment. When the error counter only is used, it is possible to process feature words into a maximum of 1024 classes at a rate of 500 words per minute.

Section 6
EXPERIMENTAL STUDY

6.1 INTRODUCTION

There are two approaches to the problem of increasing redundancy in pattern recognition algorithms:

(1) Modification of the data presented to the algorithm
(2) Revision of the basic algorithm

The experiments described in this section seek to:

- Investigate the different methods of increasing redundancy
- Establish the usefulness and limitations of each method
- Discover which of these methods is the most useful for devising an algorithm that will have maximum generalization ability but at the same time will be subject to a minimum of related disadvantages

After preliminary data were obtained to establish a frame of reference for the investigation, several series of experiments were conducted to determine the effect of:

- Increasing the number of feature words with 10-percent noise in the organizing set
- Increasing the amount of noise in an organizing set of 96 feature words
- Adding most noise to the left most bits and least noise to the right most bits
- Adding least noise to the left most bits and most noise to the right most bits
- Arranging the organizing set of 96 words with 10-percent noise in two different random sequences
- Revising the original algorithm by adding "correlation initial weights"

6.2 PROCEDURES

The synthesis algorithm described in Section 4 is the basic algorithm used in the following experimental study. As noted in Section 4.4, this algorithm tends to use as few weights as possible to achieve separation of the organizing set. For certain types of feature word spaces, this characteristic of the algorithm is unsatisfactory.*

To test the algorithm on such feature word spaces and to find ways to force the algorithm to give better results, i.e., to obtain better generalization abilities, a feature word space was devised on the basis of eight "ideal" feature words (Fig. 6-1). Arbitrary bit configurations could have been set up as the ideal words; however, use of the character configurations makes it easier to interpret the distribution of weights.

Various amounts of noise were added to the ideal feature words to form a feature word space having eight groups of closely connected points. Such a feature word space was shown geometrically in Fig. 2-8.

Noise was added to the ideal feature words by examining each bit of the word and complementing the bit with a probability of p and leaving the bit unchanged with a probability of $(1 - p)$. Sixteen-hundred samples of noisy characters were generated with $p = 0.1$, 0.2, 0.3, and 0.4.

Experiments consisted of using a particular algorithm and an organizing set of noisy feature words to find weights and thresholds, and then using these parameters on the PIP to obtain recognition values for a test set of noisy feature words. Although the experiments described in the next sections are concerned with one particular algorithm, most of the conclusions apply to any algorithm for threshold element networks which perform pattern recognition.

---

*The investigations carried out in this section emphasize the deficiencies of minimal weight algorithms. But note that the basic algorithm described in Section 4 and in Ref. 6, has been successfully applied to various problems (Ref. 4).

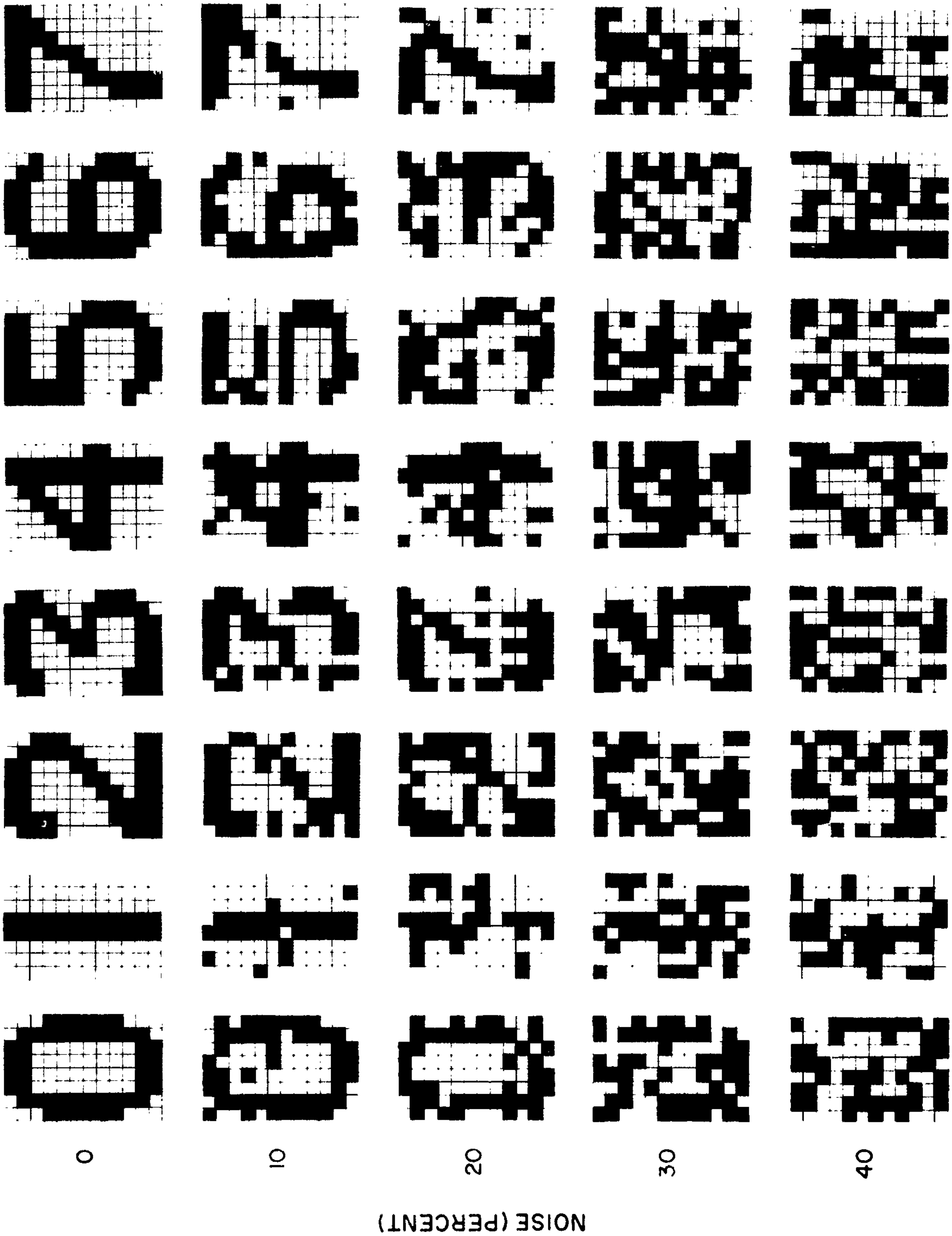


Fig. 6-1 "Ideal" Feature Words with Various Amounts of Noise Added

6.3 NOISE FREE ORGANIZING SET AND THE MINIMUM REDUNDANCY ALGORITHM

To establish a frame of reference for the projected investigation, a preliminary experiment was conducted to determine what hyperplanes would be established by the algorithm if it were presented with eight ideal feature words in a noise-free organizing set. The weight distribution obtained for 0-percent noise is shown in Fig. 6-2. Note how few weights were needed to separate the organizing set. When noisy feature words were tested against this set of weights, poor recognition resulted (see Fig. 6-9). This was because the algorithm did not use the redundant information in the feature words to formulate a decision rule that would be insensitive to the addition of noise in the feature words.

6.4 METHODS OF INCREASING REDUNDANCY BY MODIFICATION OF INPUT DATA

Redundancy can be increased in the decision rule by adding noise to the feature words. It is possible to add noise:

- By increasing the organizing set size, while keeping the amounts of noise in the feature words constant
- By increasing the amounts of noise in the feature words, while keeping the organizing set size constant

6.4.1 Increased Size of the Organizing Set

The following series of experiments tested the effect of increasing the number of feature words with 10-percent noise in the organizing set on the redundancy forced into the decision rule.

In the first experiment, the algorithm was presented with 96 samples of characters with 10-percent noise. Figure 6-2 shows the weight distribution obtained in this experiment. In the next experiment, when noisy feature words were tested against this set of weights, the recognition rate was again very low (Fig. 6-3). A study of the distribution of weights, recognition rate, the average number of non-zero weights per device, the total number of weights, and the total number of devices shown in

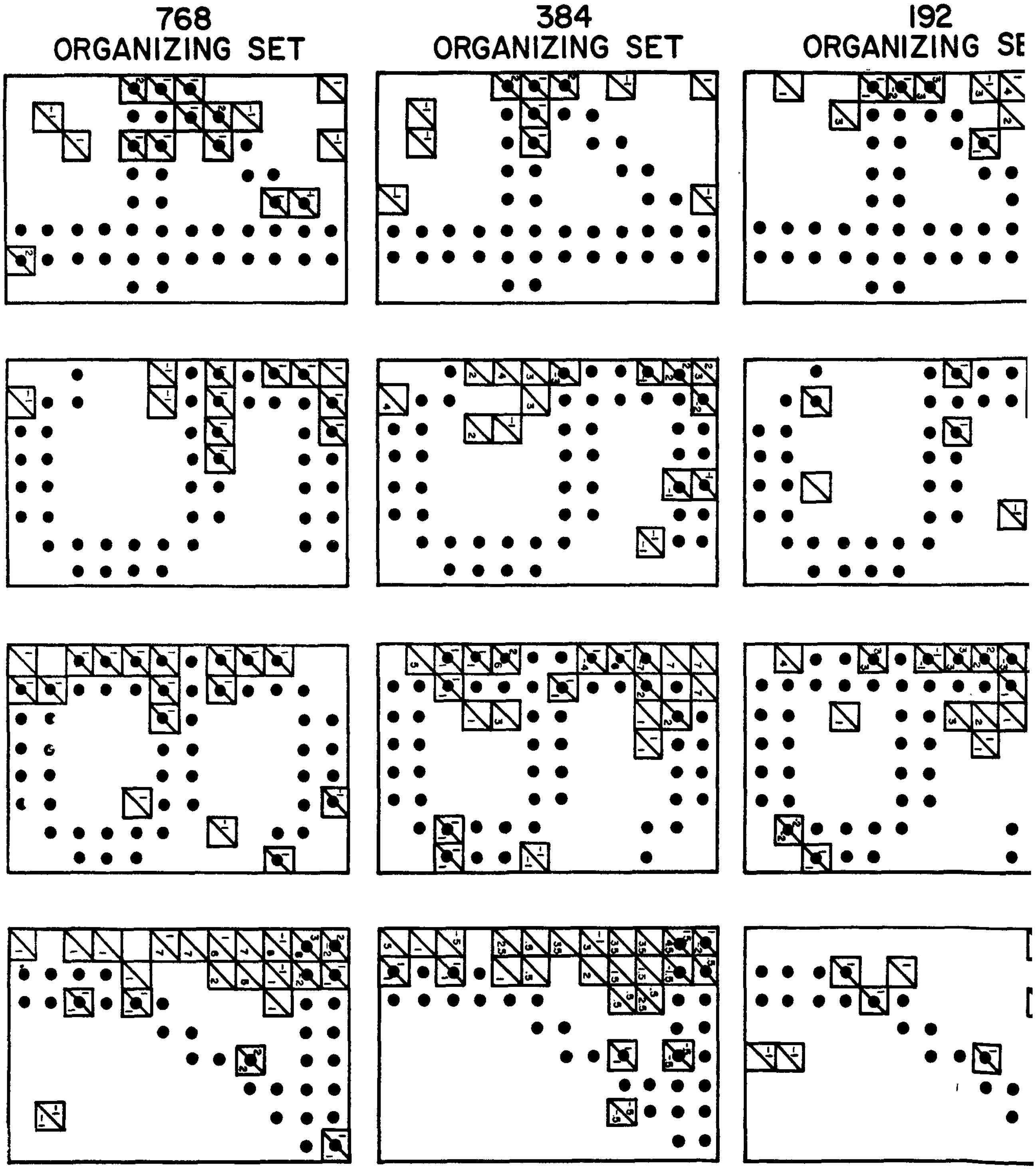


Fig. 6-

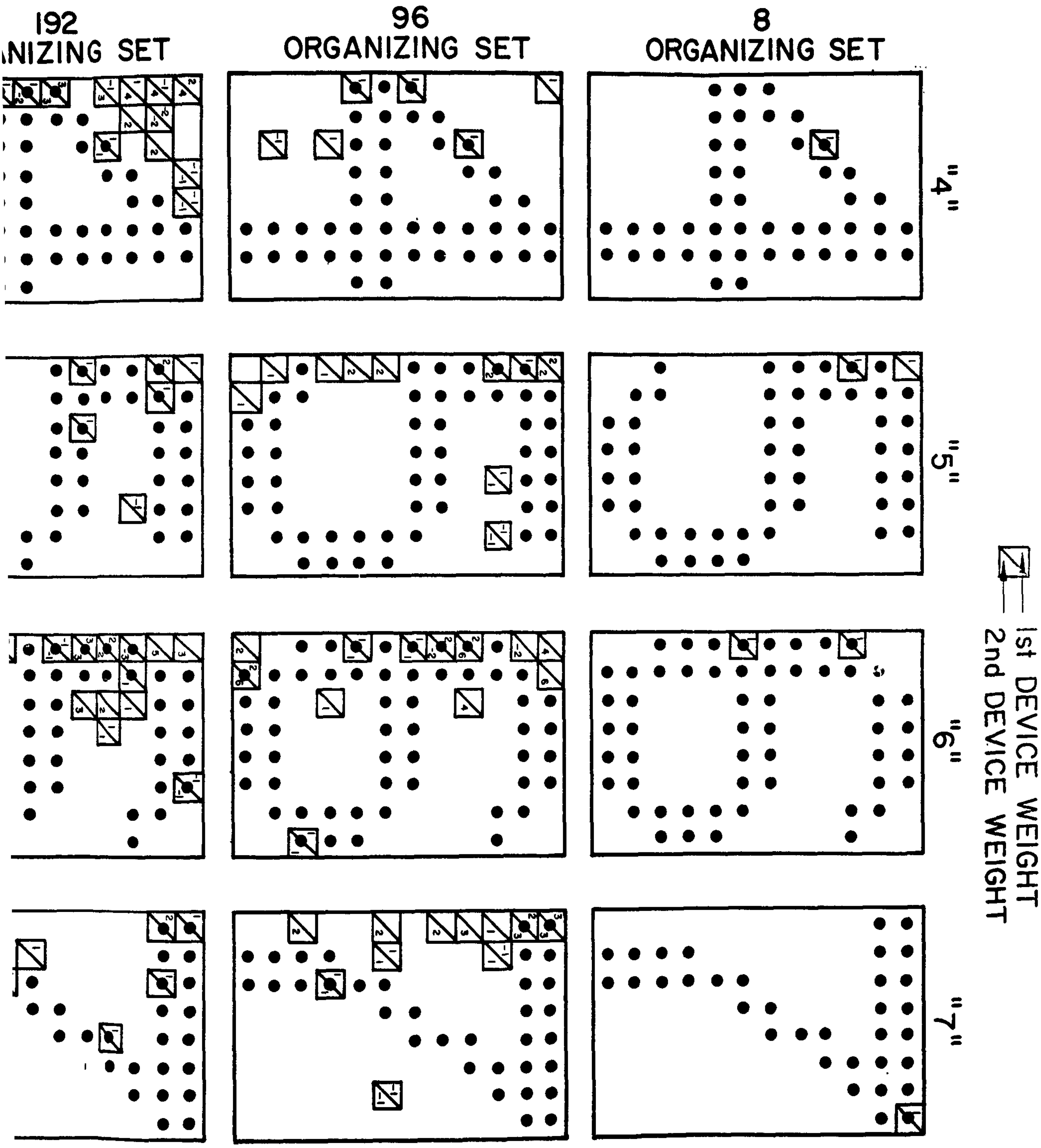


Fig. 6-2 Variation of Weight Distribution With Organizing Set Size for 10-Percent Noise

Figs. 6-2 through 6-6 reveals the effect of increasing the organizing set sizes from 96 to 768 samples. Note that the total number of devices remained nearly constant, while the average number of non-zero weights per device (which is a measure of the redundancy in the decision network) increased as the organizing set size increased. The recognition rate had a corresponding increase, and the distribution of weights in the feature word tended to outline the ideal characters as the organizing set size was increased.

The manner in which the feature words for this experiment were constructed made the feature space have eight separate groups of closely connected points. It is therefore of interest to see how the members of each class correlate with each other when they are weighted with the set of weights determined by the algorithm. An ideal situation would be to have each "0"-feature word have a weighted sum, $\Sigma\ w_i x_i - T$, approximately the same as every other "0"-feature word. Similarly, members of the other groups of feature words would have a different weighted sum; but within a class, the sums should be nearly the same.

The distribution of weighted sums for organization on the "5"-, "6"-, and "7"-feature words with 768 10-percent feature words in the organizing set is shown in Fig. 6-7. In this figure, the vertical axis represents the sum of weights minus the threshold, and the horiziontal axis is partitioned to represent the eight classes. Points are used to indicate feature words. Distance of a point above or below the horizontal line indicates distance of the feature word above or below the threshold. Each plot corresponds to the distribution for a single weight vector. Note how the sums for the various classes tend to cluster together.

6.4.2 Increased Noise In The Organizing Set

The second series of experiments tested the effect of increasing the amount of noise in the feature words in the organizing set while keeping the size of the organizing set constant at 96 feature words. Figures 6-8 through 6-13 show the effect on the distribution of weights, the recognition rate, and average number of non-zero weights per

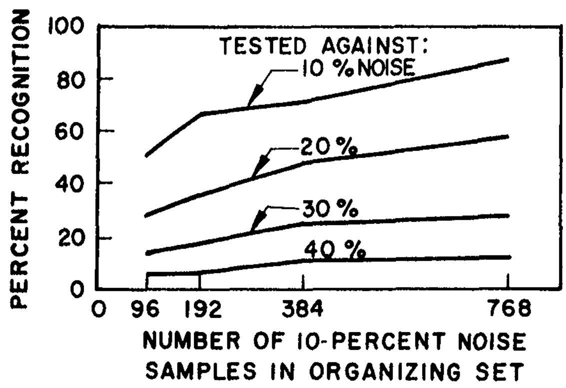


Fig. 6-3 Recognition Rate Vs. Organization Set Size

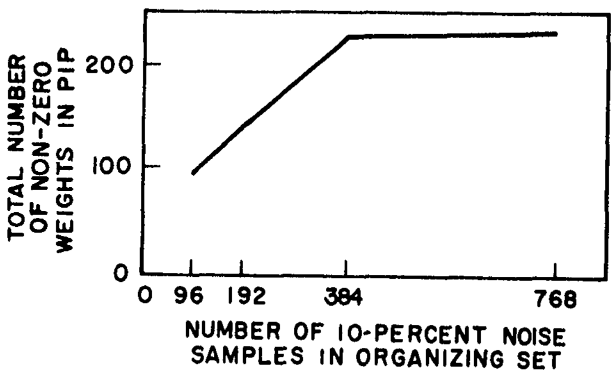


Fig. 6-5 Total Number of Non-Zero Weights Vs. Organization Set Size

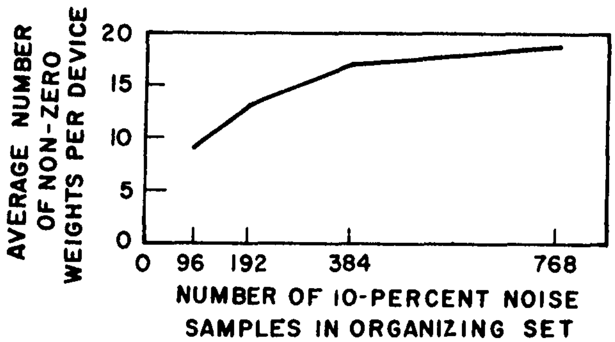


Fig. 6-4 Average Number of Non-Zero Weights Vs. Organization Set Size

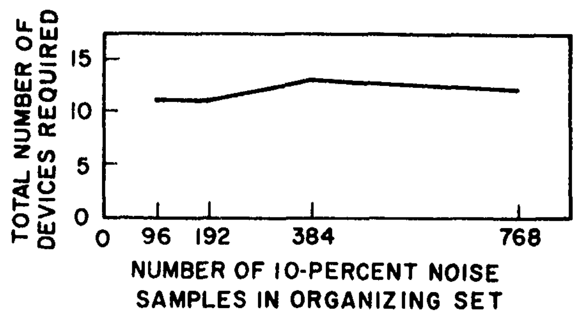


Fig. 6-6 Total Number of Devices Vs. Organizing Set Size

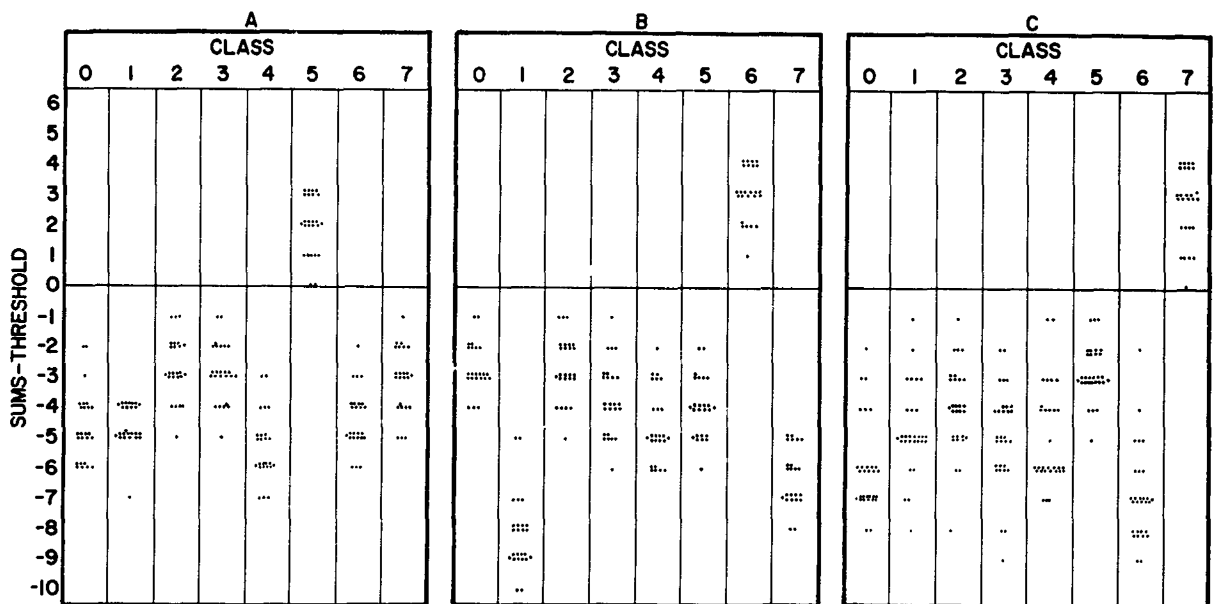
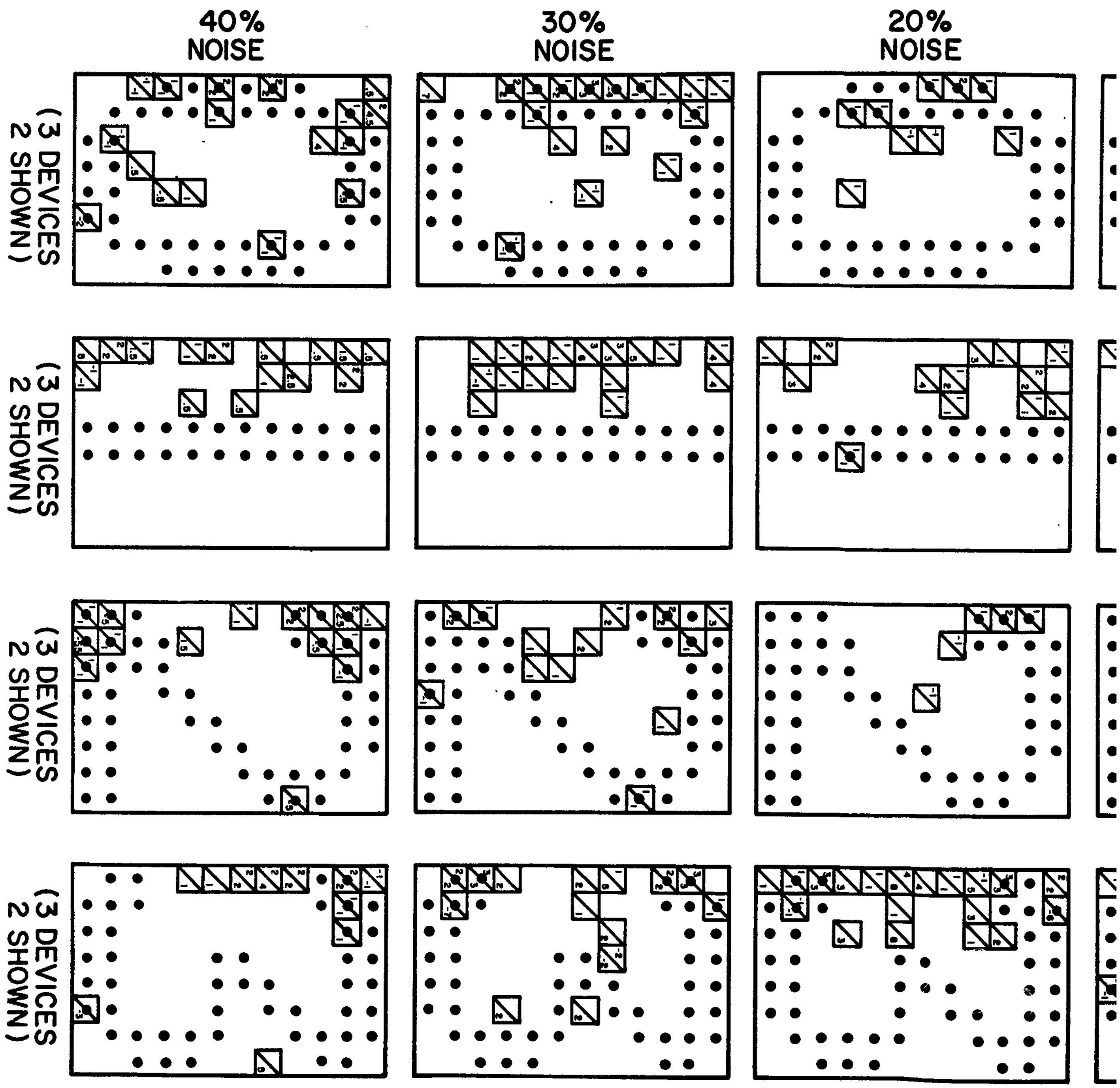


Fig. 6-7 Sum of Weights Distribution for Feature Word Groups Organized on 768 Samples of 10-Percent Noise


40% NOISE
30% NOISE
20% NOISE
(3 DEVICES 2 SHOWN)
(3 DEVICES 2 SHOWN)
(3 DEVICES 2 SHOWN)
(3 DEVICES 2 SHOWN)

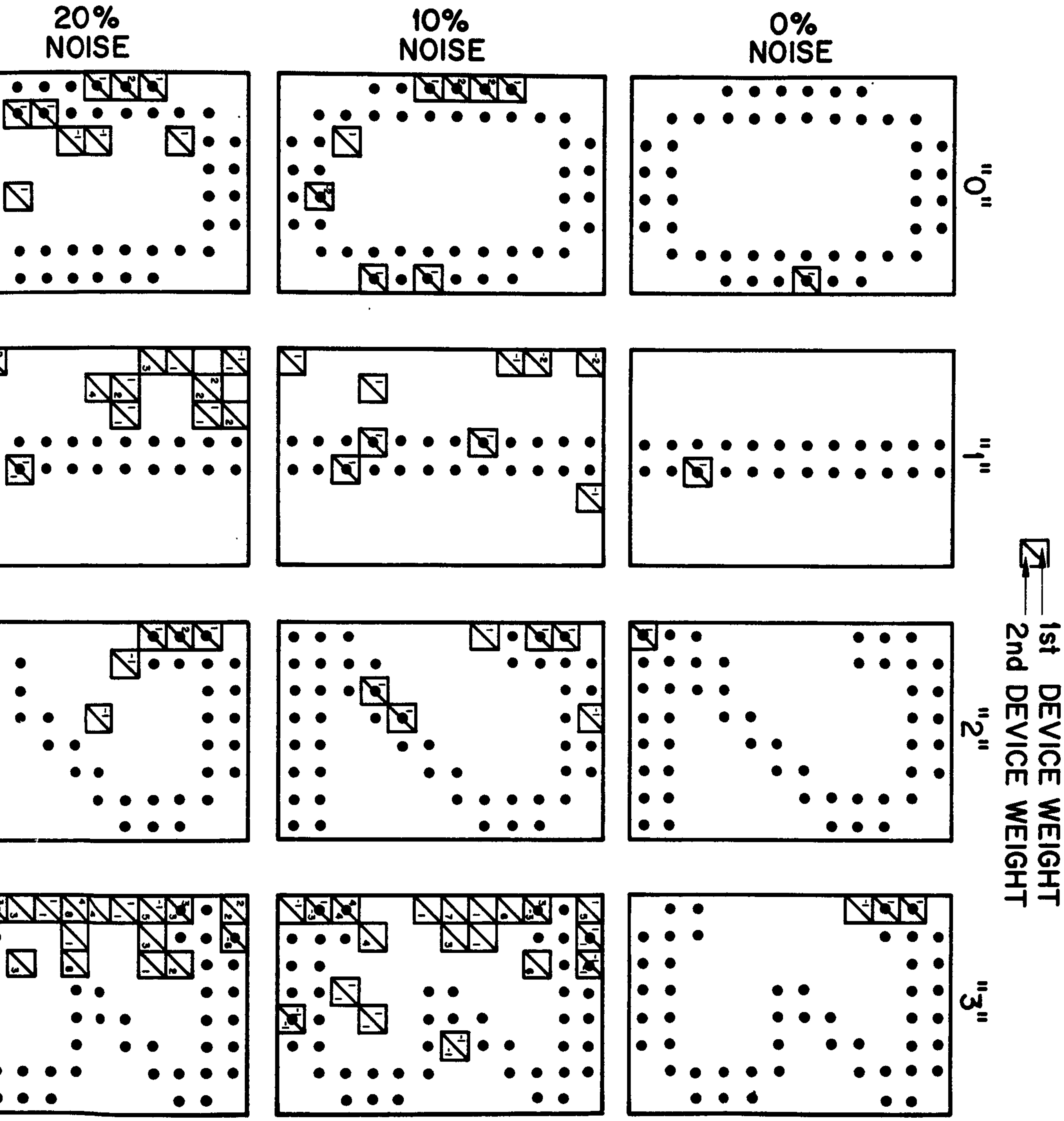


Fig. 6-8 Variation of Weight Distribution with Percent Noise for 96 Word Organizing Set

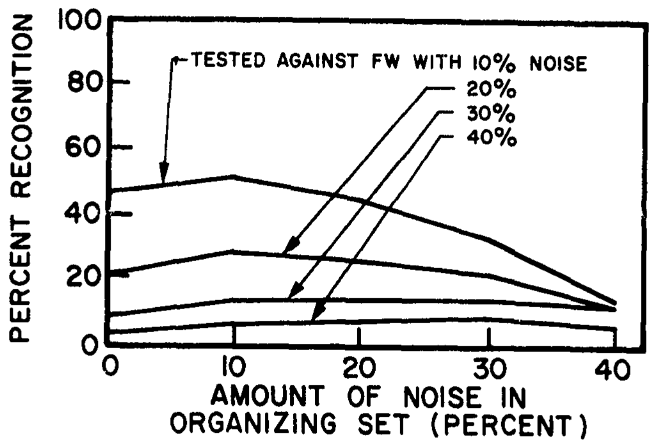


Fig. 6-9 Recognition Rate Versus Amount of Noise in Organizing Set

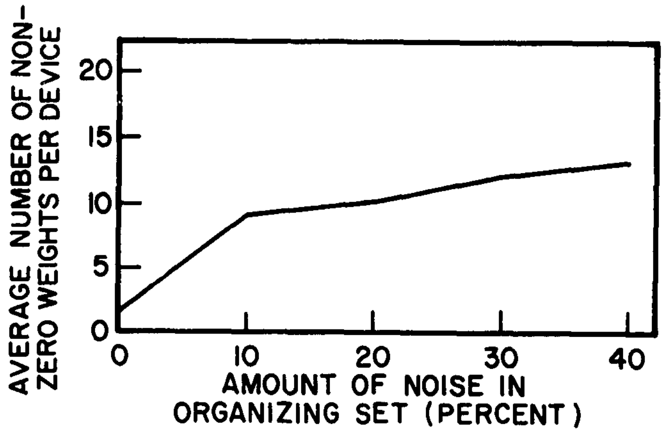


Fig. 6-10 Average Number of Non-Zero Weights Versus Amount of Noise in the Organizing Set

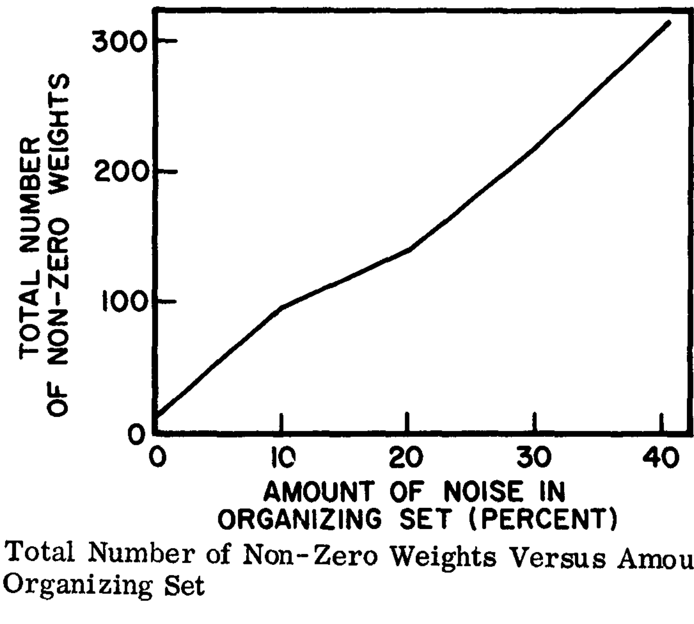


Fig. 6-11 Total Number of Non-Zero Weights Versus Amount of Noise in the Organizing Set

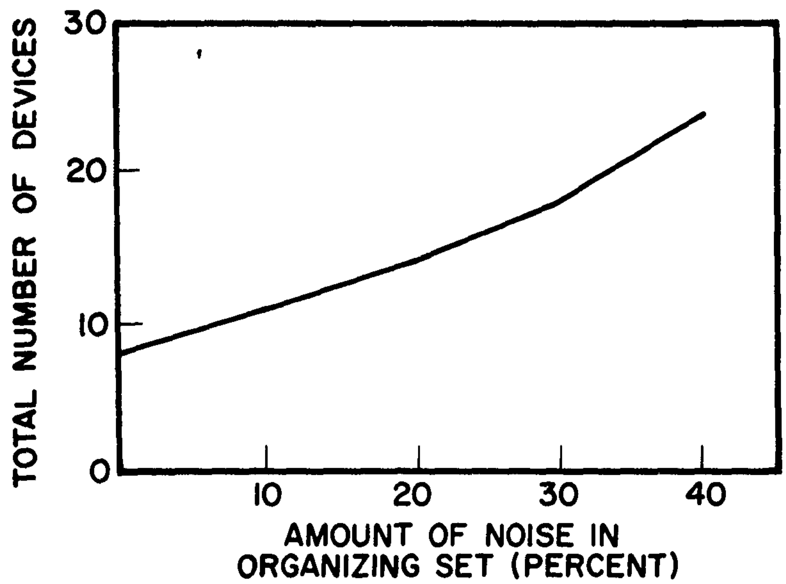


Fig. 6-12 Total Number Devices Versus Amount of Noise is Organizing Set

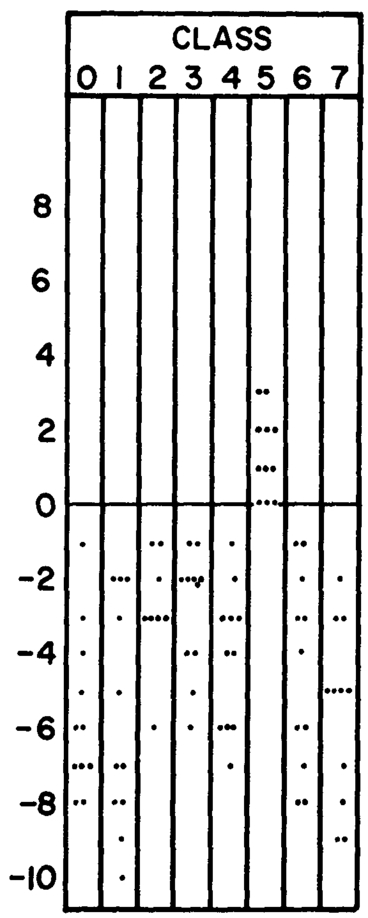


Fig. 6-13 Sum of Weights Distribution for Feature Word Groups
Organized on 96 Samples of 30-Percent Noise

device, the total number of weights, and the total number of devices, as a function of the noise in the organizing set. Note that in this experiment the total number of devices needed to separate the organizing set increased as the amount of noise was increased, which indicates that the algorithm had an increasingly difficult task of separating the organizing set. The weight distribution shows that though the average number of weights per device increased, these weights did not tend to outline the ideal characters. This weight distribution shows that the algorithm preferred to organize on the noise in the feature words rather than on the significant characteristics – a fact reflected in the decrease in recognition rate as a function of increased noise in the organizing set.

The distribution of weighted sums, show that for 30-percent noise in the organizing set the eight different classes did not correlate closely with each other (Fig. 6-13). Thus, the weight vectors produced by the algorithm did not effectively weight the significant characteristics.

6.5 CHARACTERISTICS OF THE MINIMUM REDUNDANCY ALGORITHM

6.5.1 Bit-Position Sensitivity

In the series of experiments discussed in Subsections 6.4.1 and 6.4.2, it was noted that there was a definite tendency of the algorithm to put most of the weights in the left-most bits of the feature word. Another series was therefore conducted to see if adding large amounts of noise to the left-most bits and small amounts of noise to the right-most bits of the feature word would cause the algorithm to shift the weight distribution to the right-most bit positions of the feature word. For this experiment, noise was introduced into the feature words at the rate of 45-percent noise in the left-most bit position, decreasing linearly to 5-percent noise in the right-most bit position. The weights obtained from such an organizing set were tested against the feature words with nonuniform noise, 45- to 5-percent and against feature words with uniform noise.

To further show the effect of the bit position sensitivity in the algorithm, feature words were constructed with the least amount of noise in the left-most bits and the most noise in the right-most bits. The weight distribution, using both types of feature words as an

organizing set shows that the algorithm developed weights in the right-hand portion of feature word when the left-hand portion contained 45-percent noise (Figs. 6-14 and 6-16). However, the weighted bits at the left-hand end of the feature word were still significant and lowered the overall recognition rate (Fig. 6-15). This bit-position sensitivity is a characteristic of algorithms constructed in the same manner as the minimum redundancy algorithm used in these experiments. If feature words are used for which the relative importance of the bit position is not known in advance, then the position sensitivity would be a deficiency of the algorithm.

Figures 6-15 and 6-17 show the recognition rates for the two types of feature words. Note that a significant improvement in recognition can be obtained by placing the most significant bit positions at the left end of the feature word.

6.5.2 Sequence-Dependency

It was noted in Section 4.4 that the minimum redundancy algorithm could be dependent on the order of arrangement of the feature words in the organizing set. To determine the effect of sequence in the organizing set, the organizing set of 96 words with 10-percent noise was arranged in two different random sequences. The change in recognition rate due to a change in the order of the feature words in the organizing set is given in Fig. 6-18. The two sequences differed in recognition rate from 12 percent (when tested against feature words with 10-percent noise) to 2 percent (when tested against feature words with 40-percent noise). The two curves have the same general form and shape; but in both cases, the recognition rate is less than the rate expected with optimum hyperplanes

6.6 REVISION OF THE ALGORITHM TO INCREASE REDUNDANCY BY STARTING WITH NON-ZERO WEIGHTS

To obtain redundant weights without resorting to very large organizing sets or noise perturbing methods, the original algorithm was revised by adding "correlation initial

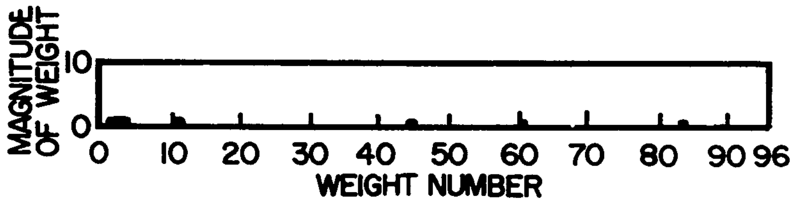


Fig. 6-14 Distribution of Weights for the Ideal "2" with Nonuniform Noise (45 Percent to 5 Percent) in the Organizing Set

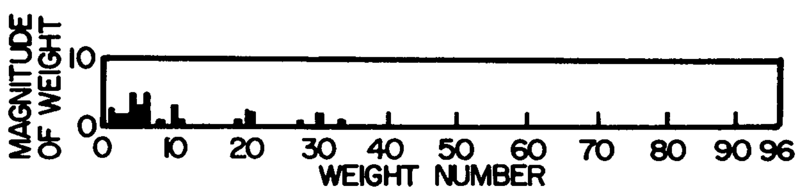


Fig. 6-16 Distribution of Weights for the Ideal "2" with Nonuniform Noise (5 Percent to 45 Percent) in the Organizing Set

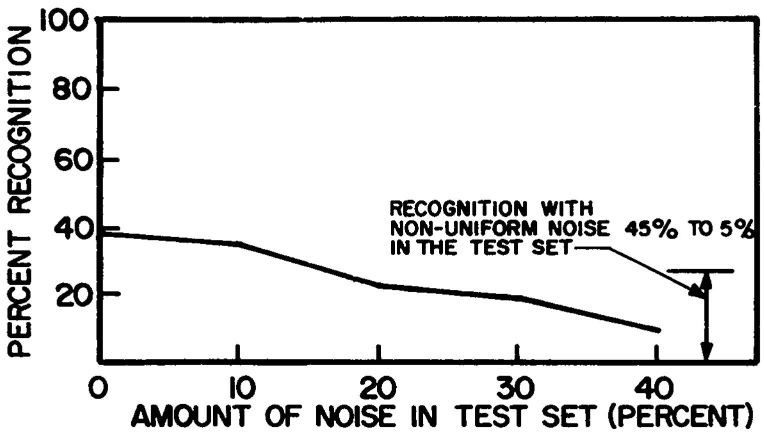


Fig. 6-15 Recognition Rate Based on an Organizing Set of 96 Feature Words with Nonuniform Noise, 45 Percent to 5 Percent

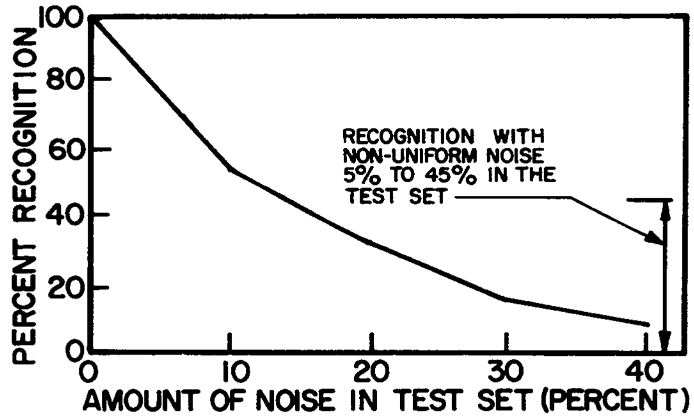


Fig. 6-17 Recognition Rate Based on an Organizing Set of 96 Feature Words with Nonuniform Noise, 5 Percent to 45 Percent

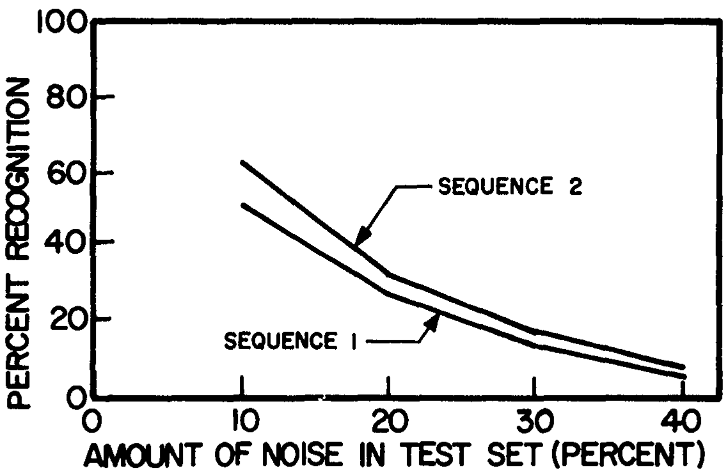


Fig. 6-18 Changes in Recognition Rate Due to the Sequence Dependency of the Algorithm

weights" to correspond with the MTCRW method of Section 3.2. This revision was accomplished by starting the original algorithm with non-zero weights, where each weight, w_i, is set to the value

w_i = (Number of "1s") – (Number of "0s") in column x_i for 1-mapped feature words.

The "correlation initial weights" thus obtained are used as the starting weights for the algorithm; all the procedures of the algorithm then were used as described in Section 3.2.

With the addition of the "correlation initial weights," the algorithm becomes less sequence dependent and less bit-position sensitive. Since the initial weights are calculated independently, there can be no bit position sensitivity in the initial weights. Also, with initial weights, all feature words have an initial sum calculated with these weights. Because these sums determine the order in which the algorithm looks at feature words, the original order of these feature words becomes less important.

Figure 6-19 compares the recognition rate obtained with the basic algorithm using 768 feature words as an organizing set and the basic algorithm with "initial correlated weights" using 96 feature words as an organizing set. Both organizing sets contained feature words with 10-percent noise. The two curves are similar in shape, but the recognition with initial correlated weights is always higher, even though fewer samples were used in the organizing set. This demonstrates the tradeoff between using:

- The basic algorithm and a larger organizing set
- The initial correlated weights and a small organizing set

Figure 6-20 shows that the recognition rate is not greatly improved by increasing the organizing set size when initial correlated weights are used. The improvement in recognition in relation to the organizing set size of the basic algorithm is shown for comparison.

Figure 6-21 shows the relationship between the recognition rate and the amount of noise in the organizing set when initial correlated weights are used. Note that the final decision rule determined by the algorithm achieves higher recognition when tested against the type of data in the organizing set. This is because the threshold value is different for different amounts of noise. Consequently, a value that has high recognition for 10-percent noise will not have high recognition on 40-percent noise and visa versa. The final recognition obtained using initial correlated weights was higher in every case than when the basic algorithm was used (Fig. 6-9).

To illustrate how the bit-position sensitivity of the basic algorithm is improved by the addition of initial correlated weights, the experiment using nonuniform noise reported in Section 6.6 was repeated using the initial weights. Figures 6-22 through 6-25 show the results. Note that with the initial correlated weights, the largest magnitude weights were assigned to the bit positions with the least noise, and the lowest magnitude weights were assigned to the bit positions with the most noise. Nevertheless, the bit-position sensitivity of the basic algorithm can still be observed in this experiment. The recognition for feature words where left-most bits contained the least noise was higher than recognition for feature words whose left-most bits

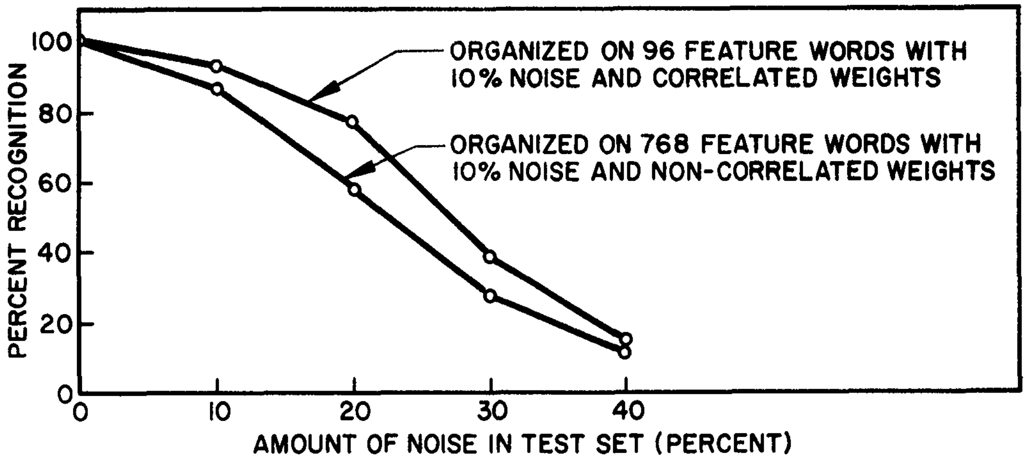


Fig. 6-19 Comparison of Recognition Rate between the Best Noncorrelated Algorithm and the Correlated Algorithm

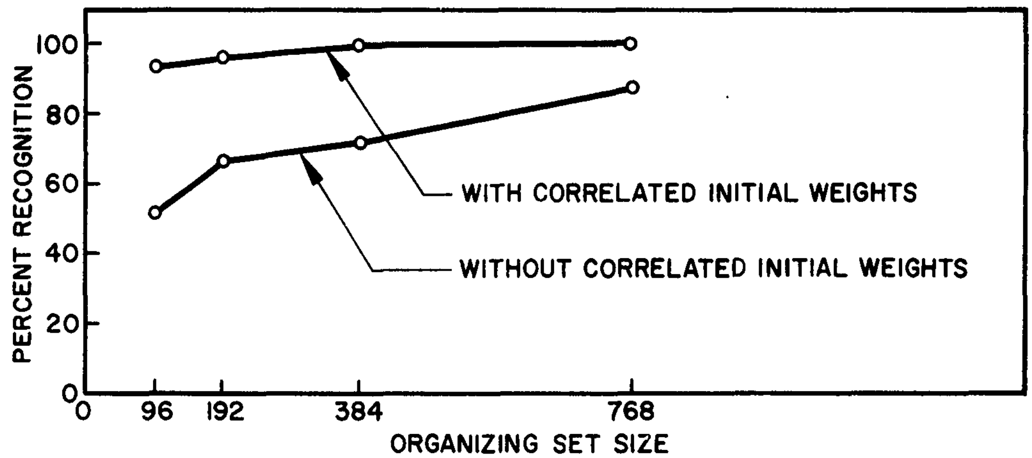


Fig. 6-20 Comparison of Recognition Versus Organizing Set Size for Correlation and Noncorrelation

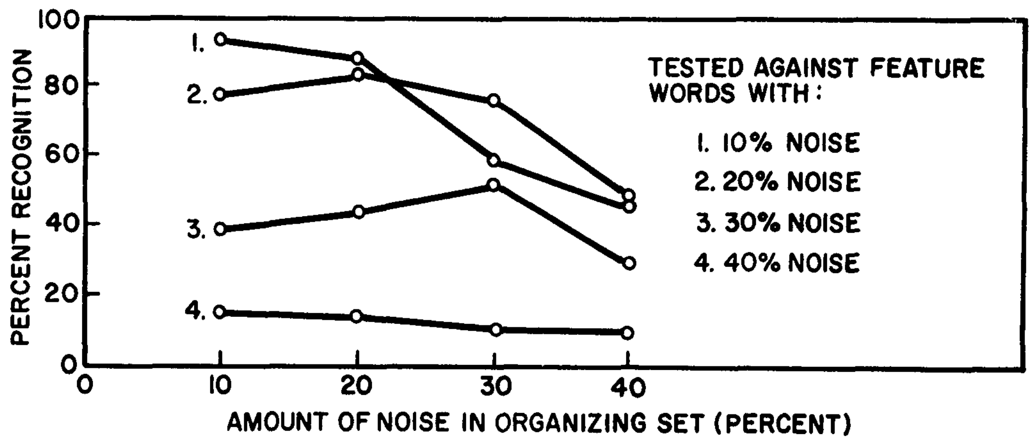


Fig. 6-21 Recognition Rate Versus Amount of Noise in the Organizing Set

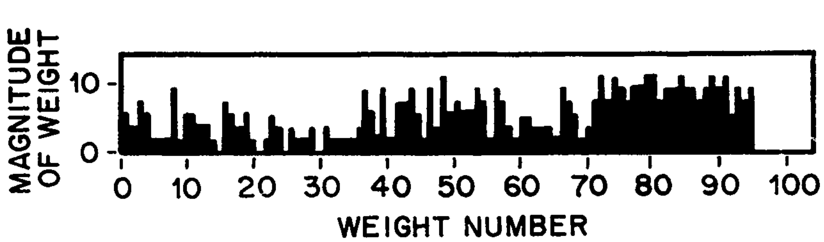


Fig. 6-22 Distribution of Weights for the Ideal "2" with Nonuniform Noise (45 Percent to 5 Percent) in the Organizing Set

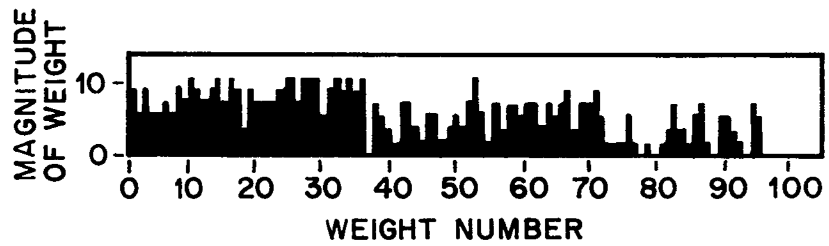


Fig. 3-24 Distribution of Weights for the Ideal "2" with Nonuniform Noise (5 Percent to 45 Percent) in the Organizing Set

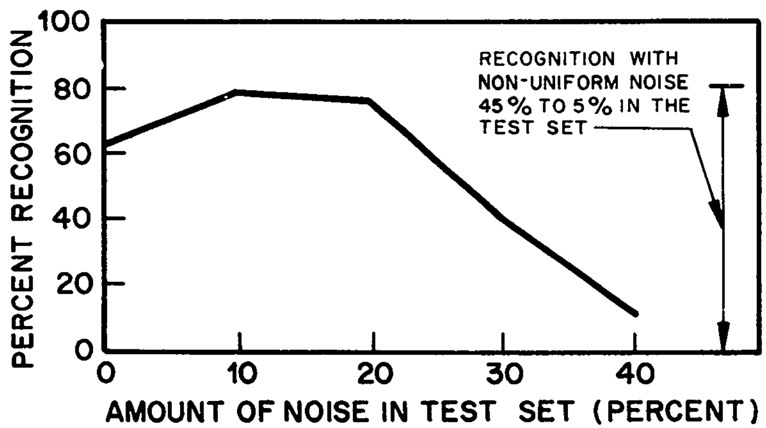


Fig. 6-23 Recognition Rate Based on an Organizing Set of 96 Feature Words with Nonuniform Noise (45 Percent to 5 Percent)

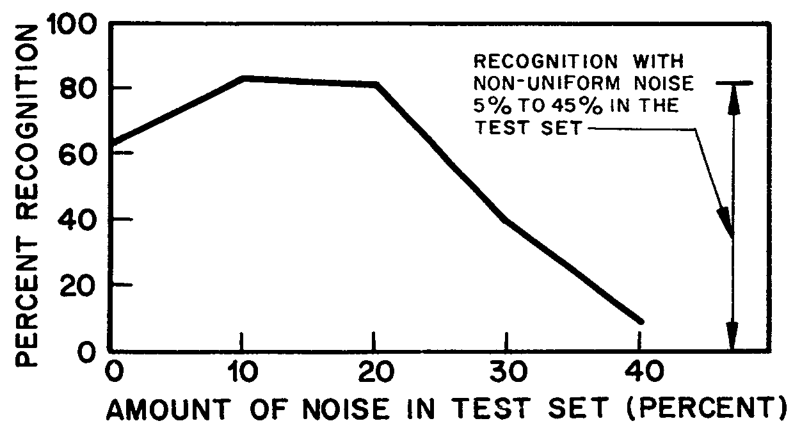


Fig. 6-25 Recognition Rate Based on an Organizing Set of 96 Feature Words with Nonuniform Noise (5 Percent to 45 Percent)

contained the most noise. The increase was from 80 to 83 percent. These recognition rates were significantly higher than those that could be obtained with the basic algorithm (see Figs. 6-14 through 6-17).

6.7 SUMMARY OF RESULTS

The series of experiments presented in this section yielded the following results:

- A synthesis algorithm which attempts to use as few weights as possible for pattern recognition (1) may have mechanization advantages; (2) does not lead to networks with good generalization abilities for certain types of feature word spaces.
- The generalization abilities of the basic algorithm can be improved by adding initial correlated weights or by using very large organizing set sizes.
- There was a definite tendency of the basic algorithm to put most of the weights in the left-most bits of the feature word.
- A marked improvement in recognition was obtained by placing the most significant bit positions at the left end of the feature word.

With the addition of "correlation initial weights":

- The algorithm became less sequence dependent and less bit-position sensitive.
- The size of the organizing set had relatively little effect on the recognition rate.

The results of the experiments show that for the feature-word space consisting of ideal characters with additive noise, the best results were obtained with an algorithm which used initial correlated weights.

Section 7
FUTURE WORK

The investigations presented in this report will be continued to determine the effect of other variations of the synthesis algorithm on the recognition of other feature-word spaces. The first investigation will consider feature word spaces with more than one "ideal" for each recognition class; i.e., four different "0's", four different "1's," ..., four different "7's" will be considered. Several different ways of computing correlated initial weights will be used to determine which is best for this type of feature-word space.

The second investigation will be a continuation of the feature-word construction procedures given in Ref. 4. In these investigations, the linear coding will be applied to actual line-crossing information in hand-printed numerical characters. The various forms of the synthesis algorithm will then be used to process this type of feature word space. As another part of this investigation, initial weights corresponding to binary numbers will be used with the basic algorithm to determine the generalization abilities of such an algorithm as applied to binary coded data.

The third investigation will consider methods of automatic subclass determination. It will deal specifically with the problem of patterns that consist of considerably different configurations, but are still considered members of the same class: e.g., capital A, lower case a, and script *a*. For such problems it is difficult to develop a pattern recognition algorithm which can generalize on the different feature words produced by these subclasses. If the subclasses are identified and given different desired classification codes, improved recognition will result.

Several recent papers have dealt with the problem of the automatic determination of subclasses for a representation set of feature words (Refs. 8, 10, 11, and 12). The projected investigations will test various approaches to determine the most useful in pattern recognition.

Section 8
REFERENCES

1. R. L. Mattson, "A Self-Organizing Binary System," Proc. EJCC, 1959, pp. 212-217

2. Lockheed Missiles & Space Division, An Approach to Pattern Recognition, Using Linear Threshold Devices, by R. L. Mattson, 702680, Sunnyvale, Calif., Sep 1960

3. Lockheed Missiles & Space Company, An Approach to General Pattern Recognition, by M. Fischler, R. L. Mattson, O. Firschein and L. D. Healy, 6-90-62-2, Sunnyvale, Calif., April 1962; to be presented at the IRE International Symposium on Information Theory, Sept 1962

4. -----, Feature Word Construction for Use With Pattern Recognition Algorithms: An Experimental Study, by R. L. Mattson and O. Firschein, 6-90-62-58, Sunnyvale, Calif., Aug 1962

5. R. McNaughton, "Unate Truth Functions," IRE Trans. on Electronic Computers, vol. EC-10, No. 1, Mar 1961, pp. 1-5

6. M. A. Fischler, "Investigations Concerning the Theory and Synthesis of Linearly Separable Switching Functions," Ph.D. Thesis, Stanford University, Stanford, Calif., Jun 1962

7. R. L. Mattson, "The Analysis and Synthesis of Adaptive Systems Containing Networks of Threshold Elements," Ph.D. Thesis, Stanford University, Stanford, Calif., Jun 1962

8. Lockheed Missiles & Space Company, Hyperplane Techniques in Pattern Recognition, by M. A. Fischler, 6-90-62-59, Sunnyvale, Calif., Jul 1962

9. Lockheed Missiles & Space Division, Digital Data Processor for Pattern Recognition Experiments, by L. D. Healy, 895013, Sunnyvale, Calif., Jan 1961

10. R. E. Bonner, "A "Logical Pattern" Recognition Program," <u>IBM Journal of Research</u>, Jul 1962, pp. 353-360

11. R. L. Mattson, "Application of Unger-Paull Technique to Subclass Determination," Lockheed Internal Note, Aug 1962

12. R. L. Mattson, "Application of Physical Mechanics to Subclass Determination," Lockheed Internal Note, Aug 1962